REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 19/Feb/2002	2. REPORT TYPE THESIS	3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
ROCKET PLUME TOMOGRAPHY OF COMBUSTION SPECIES

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
CAPT KUTRIEB JOSHUA M

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NAVAL POSTGRADUATE SCHOOL

8. PERFORMING ORGANIZATION REPORT NUMBER
CI02-26

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

13. SUPPLEMENTARY NOTES

14. ABSTRACT

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20020305 176

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		92	19b. TELEPHONE NUMBER *(Include area code)*

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ROCKET PLUME TOMOGRAPHY OF COMBUSTION SPECIES

by

Joshua M. Kutrieb

December 2001

Thesis Advisor:	Christopher Brophy
Co-Advisors:	Jose Sinibaldi
	Ashok Gopinath
Second Reader:	David Netzer

THIS PAGE INTENTIONALLY LEFT BLANK

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 2001	3. REPORT TYPE AND DATES COVERED Engineer's Thesis	
4. TITLE AND SUBTITLE: Rocket Plume Tomography of Combustion Species		5. FUNDING NUMBERS NF046110001006	
6. AUTHOR(S) Kutrieb, Joshua M.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) Ballistic Missile Defense Organization (BMDO), Washington DC 20301-7100 Air Force Research Laboratory (AFRL), 10 E. Saturn Blvd, Edwards AFB CA 93524-7680		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Interest in accurate detection and targeting of aggressor missiles has received considerable interest with the national priority of developing a missile defense system. Understanding the thermal signatures of the exhaust plumes of such missiles is key to accomplishing that mission. Before signature models can be precisely developed for specific rockets, the radiation of the molecular or combustion species within those plumes must be accurately predicted. A combination translation / rotation scanning diagnostic technique has been developed to map the combustion species of a rocket plume and characterize its radiation properties. Using new infrared spectrometer and fiber optic cable technology to transmit the signal spectrum of interest, the custom designed mechanism can sweep through two dimensions of a steady-state rocket exhaust. A glow bar, or blackbody simulator, is shuttered on the opposite side of the plume, allowing the spectrometer to measure both the emission and absorption spectra. This thesis demonstrated the first time use of fiber optic cable to transmit infrared emission / absorption (E/A) spectra from a rocket plume to an infrared detector. This new fiber optic configuration allows for rapid translation and rotation around the rocket plume, establishing the capability for rapid spatial characterization of the combustion species present. Experimental results may then be compared to DoD rocket plume model predictions to highlight areas for improvement.

14. SUBJECT TERMS Rocket Plume Exhaust, Spectral Imaging, Emission / Absorption, Combustion Species, Signature			15. NUMBER OF PAGES 92
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

THIS PAGE INTENTIONALLY LEFT BLANK

ROCKET PLUME TOMOGRAPHY OF COMBUSTION SPECIES

Joshua M. Kutrieb
Captain, United States Air Force
B.S., United States Air Force Academy, 1992

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING
AERONAUTICAL AND ASTRONAUTICAL ENGINEER
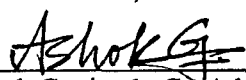
from the

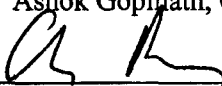## NAVAL POSTGRADUATE SCHOOL
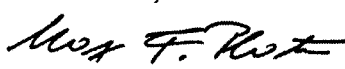**December 2001**

Author: _____
Joshua M. Kutrieb

Approved by: _____
Christopher Brophy, Thesis Advisor

_____
Jose Sinibaldi, Co-Advisor

_____
Ashok Gopinath, Co-Advisor

_____
David Netzer, Second Reader

_____
Max Platzer, Chairman
Department of Aeronautics and Astronautics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Interest in accurate detection and targeting of aggressor missiles has received considerable interest with the national priority of developing a missile defense system. Understanding the thermal signatures of the exhaust plumes of such missiles is key to accomplishing that mission. Before signature models can be precisely developed for specific rockets, the radiation of the molecular or combustion species within those plumes must be accurately predicted. A combination translation / rotation scanning diagnostic technique has been developed to map the combustion species of a rocket plume and characterize its radiation properties. Using new infrared spectrometer and fiber optic cable technology to transmit the signal spectrum of interest, the custom designed mechanism can sweep through two dimensions of a steady-state rocket exhaust. A glow bar, or blackbody simulator, is shuttered on the opposite side of the plume, allowing the spectrometer to measure both the emission and absorption spectra. This thesis demonstrated the first time use of fiber optic cable to transmit infrared emission / absorption (E/A) spectra from a rocket plume to an infrared detector. This new fiber optic configuration allows for rapid translation and rotation around the rocket plume, establishing the capability for rapid spatial characterization of the combustion species present. Experimental results may then be compared to DoD rocket plume model predictions to highlight areas for improvement.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Combustion diagnostics have generated considerable interest, from understanding the physics of the combustion processes to examining the exhaust properties from rocket plumes and turbine engines. Determining, predicting, and modeling the spectral signatures of these devices, especially those devices operating as aggressive weapons against the United States and its allies, is considered a national priority. Understanding those spectral signatures requires detailed knowledge of the radiation emitted from and absorbed by the molecular species within the plumes, or the combustion species. From the work of early pioneers such as Dr. Charles Ludwig who measured the correlation of emission and absorption or transmittance from these combustion species to their concentration and temperature for NASA, the species of interest can be characterized through computational programs.[3]

A translation / rotation scanning diagnostic technique has been developed to rapidly map the combustion species of a rocket plume and characterize its radiation properties. This thesis involved two major efforts: custom hardware design and computer code development, followed by their subsequent integration. The Naval Postgraduate School's Rocket Propulsion and Combustion Lab provides a unique facility where high fidelity diagnostic testing can be conducted on small, but true rocket exhaust plumes, lessening the fear that a test anomaly would destroy the diagnostic equipment. The hardware goal was to design a setup that could rapidly translate a horizontally opposed infrared collection assembly and blackbody source linearly through a 1.0-inch diameter rocket exhaust plume, rotate the entire mechanism a specified angle, then repeat this procedure multiple times, maximizing data collection during the rocket engine firing. At each translation line of sight (LOS) or "zone," emission and absorption measurements were taken by operating a shutter between the plume and blackbody. An open shutter provides plume absorption or transmission data while a closed shutter provides emission only. Since a bulky infrared spectrometer cannot be easily translated and rotated around an exhaust plume without compromising its reliable operation (the blackbody and shutter hardware are more robust and smaller), fiber optic cable transmitted the signal from the plume to the stationary infrared spectrometer. This thesis demonstrated the first time use

1

of fiber optic cable to transmit emission / absorption spectra from a rocket plume to an infrared spectrometer.

The emission and absorption information was collected and stored as ASCII files, then input into a combustion species deconvolution code, provided to us by the Arnold Engineering Development Center (AEDC) at Arnold Air Force Base (AFB), Tennessee. The AEDC deconvolution code, given emission and transmission information for a particular combustion species, will calculate concentration and temperature of that species while iterating with the Standard Infrared Radiation Model (SIRRM). Due to factors beyond our control, the emission and absorption data collected for this research was not sufficient to allow the deconvolution code to converge (see Section IV). If combustion species concentrations and temperatures are calculated, results may then be compared to DoD rocket plume model predictions to highlight areas for improvement.

II. EXPERIMENTAL APPROACH

A. RADIATION INTENSITY MEASUREMENT TECHNIQUE FROM A GASEOUS MEDIUM

The purpose of this thesis was to develop a portable diagnostic apparatus to determine static temperature and molecular species concentration profiles in a nonuniform combustion flowfield, assuming nonsymmetric flow. The fundamental requirements of this goal have been achieved, with the groundwork for future research and fidelity improvements firmly established. The molecular species deconvolution numerical program used for this research was originally developed by Dr. Charles Limbaugh and Dr. L. Brewer, who were funded by the Arnold Engineering Development Center (AEDC) at Arnold Air Force Station in Tennessee. Their research translated a source (globar) and receiver unit (monochromator) across one dimension of a Titan Liquid Rocket Engine exhaust plume in the early 1970s (see Figure 1). Plume diameter was approximately 160 cm.

——— Light Path	M	Prism monochromator
m_1	Spherical Mirror	S	2.7-cm-diam Burner
m_2	Plane Mirror	T	Traversing Mechanism
C	Chopper and Shutter	B	N_2 Purged Plexiglass Enclosures
G	Globar	s	Monochromator Entrance Slit
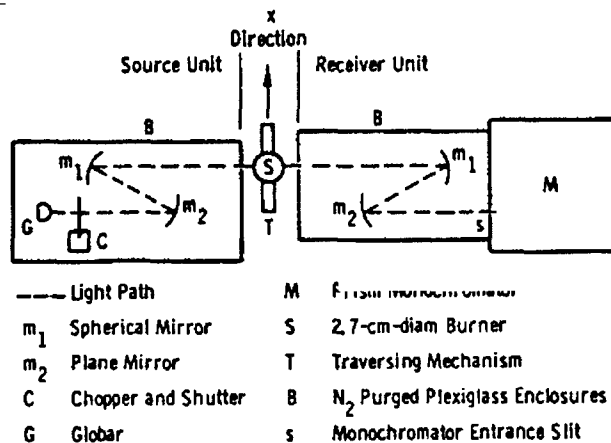
Figure 1.   AEDC Experimental Scanning Apparatus.

This thesis work has created a more elaborate setup that scans a 2-5 cm diameter rocket exhaust plume through multiple linear positions, multiple angles, and varying plume depth, using a fiber optic cable to transmit the measured signal. The spectral information collected for this research is essentially the same as conventional E/A measurements:

3

simply more data from more angles, and therefore, can use the same deconvolution code. The AEDC deconvolution code provides molecular species concentrations and temperatures given emission and absorption measurements for multiple transmission paths through the plume at a particular plume depth. Assuming a particular wavelength band of significant intensity, this data can produce a 2-dimensional, symmetrical image of the species' properties. A detailed discussion of the code is found in Section III.C. Additional data manipulation added for this thesis produces a 2-dimensional image of species properties, using information compiled from different translation angles. The following text describes the theory of the original AEDC work, as taken directly from Dr.'s Limbaugh and Brewer's paper "Infrared Band Model Technique for Combustion Diagnostics."[3]

The use of models to represent individual or composite bands of molecular species (in this case, H_2O, CH_4, CO_2, and CO) is necessary because of the large number of lines that are overlapping or superimposed. Band models provide a mathematical means of averaging over fine structure, without distorting the band contour, by assuming combinations of randomly or evenly spaced lines with a constant or varying distribution of intensities. This is done in such a way that the mathematical average values of radiance and transmittance are consistent with the physical averages over a wavelength interval. Below are the expressions for radiance and transmissivity using the band model, which is formulated for the homogeneous case, and the Curtis-Godson approximation, which allows extension from the homogeneous to the inhomogeneous case.

The equation for radiance N from a gas is

$$N = \sum_i N_B(T_i)(\tau_{i-1} - \tau_i) \qquad (1)$$

$N_B(T_i)$ = Planck blackbody radiance at temperature T_i and at a particular frequency

τ_i = transmissivity of zones $1,2,\ldots,i$

τ_{i-1} = transmissivity of zones $1,2,\ldots,i\text{-}1$

The equation for transmissivity of a single zone of constant temperature and pressure, derived using the random band model with constant line widths and a delta-function distribution for line strengths is

$$-\ln\tau = 2\pi(\gamma/d)f(x) \tag{2}$$

where

$$x = [(S/d)l/2\pi(\gamma/d)] \tag{3}$$

Use of the Curtis-Godson approximation allows an immediate extension of Eqn (2) to the many-zone (inhomogeneous) case:

$$-\ln\tau = \frac{\sum_h x_h \left[\dfrac{-\ln\tau_h}{f(x_h)}\right]^2}{\sum_h x_h \dfrac{-\ln\tau_h}{f(x_h)}} f\left(\frac{\left[\sum_h x_h \dfrac{-\ln\tau_h}{f(x_h)}\right]^2}{\sum_h x_h \left[\dfrac{-\ln\tau_h}{f(x_h)}\right]^2}\right) \tag{4}$$

The definitions of the symbols in Eqns (2-4) are

τ = transmissivity

γ = line half-width at half-height, cm

S = line strength

d = line spacing, cm

l = zonal path length, cm

h = reference to a single zone

$f(x)$ = Ladenburg and Reiche (LR) function

The band model parameters $(\gamma/d)_h$ and $(S/d)_h$ for a particular zone are further defined as follows:

$$(\gamma/d)_h = (\gamma_a^\circ/d)_h\, p_h^a + (\gamma_b^\circ/d)_h\, p_h^b \tag{5}$$

$$(S/d)_h = (S^\circ/d)_h\, p_h^a \tag{6}$$

where

$\gamma^c/d, S^c/d$ = experimentally determined tabular functions of temperature (*Ludwig eqns)

p_h^a = pressure of the absorbing gas in zone h, atm

p_h^b = pressure of a broadening gas (s), atm

As noted previously, transmission of band radiation through combustion gases does not obey Beer's law. Therefore, the radiance N and the transmissivity τ cannot be scanned laterally outside the gas and inverted to radial positions using geometrical methods such as that provided by the Abel integral. However, eqns (1) and (4) are expressions for the radiance and transmissivity in the inhomogeneous gas in four variables, T_h and p_h and the measurable quantities N and τ, and therefore are, in principle, solvable for the T_h and p_h quantities.

Because of the nonlinear coupling of eqns (1) and (4), direct analytic solution of T_h and p_h is impossible. However, the solution is amenable to numerical techniques and solution is tractable.

To demonstrate the method of solution used here, consider the combustion source being modeled by a series of concentric cylinders with T_h and p_h constant within each cylinder (see Figure 2). The radiance and transmissivity measurements are thus the sum of the contribution from the portion of the cylinder observed by the spectrometer. The first measurements, N_a and τ_a, are the result of only zone 1; the second measurements, N_b and τ_b, are the result of both zones 1 and 2, and so on.

6
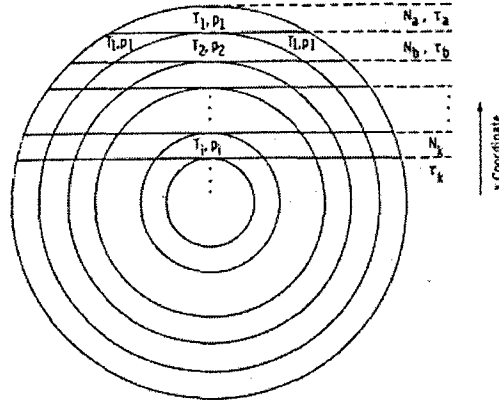
Figure 2.   Cross Sectional Schematic of Inhomogeneous Cylindrically Symmetrical Source

Showing Zones of Constant Temperature, Concentration, and Measurement Locations.

From eqns (1) and (2) and referring to Figure 2, a solution for T_1 and p_1 is immediate with N_a and τ_b. N_b and τ_b, however, include contributions from both zones 1 and 2, for which T_2 and p_2 are not known, and eqns (1) and (4) must be used. Similarly, each succeeding radiance and transmissivity measurement will include contributions from the central most zone for which T and p are not known. Thus, from Figure 2, the measurement of N_k, τ_k is a function of the unknown T_i, p_i and the known T, p from each of the other zones. This solution can be effected by considerations of the functional dependence of N and τ and application of the Newton-Raphson iterative technique.

As noted above, the only unknown parameters for each of the radiance and transmissivity measurements will be the temperature and pressure of an inner zone, T_i and p_i. For N_b and τ_b these are T_2 and p_2, etc. Hence the radiance and transmissivity can be written functionally as

$$N = N(T_i, p_i), \qquad \tau = \tau(T_i, p_i) \tag{7}$$

The differential of this system with respect to these variables is

$$dN = (\partial N / \partial T_i) dT_i + (\partial N / \partial p_i) dp_i$$

$$d\tau = (\partial \tau / \partial T_i) dT_i + (\partial \tau / \partial p_i) dp_i \tag{8}$$

7

where eqns (1) and (4) are used to obtain the partial derivatives.

Now, let some T_i, p_i be assumed. Call these $T_i^{(0)}$, $p_i^{(0)}$ and use these to calculate radiance, transmittance, and the partial derivatives in the system, eqns (7) and (8), or

$$N^{(0)} = N[T_i^{(0)}, p_i^{(0)}],$$

$$\tau^{(0)} = \tau[T_i^{(0)}, p_i^{(0)}],$$

$$N_x - N_0 = \left.\frac{\partial N}{\partial T_i}\right|_0 (T_i^{(1)} - T_i^{(0)}) + \left.\frac{\partial N}{\partial p_i}\right|_0 (p_i^{(1)} - p_i^{(0)}),$$

$$\tau_x - \tau_0 = \left.\frac{\partial \tau}{\partial T_i}\right|_0 (T_i^{(1)} - T_i^{(0)}) + \left.\frac{\partial \tau}{\partial p_i}\right|_0 (p_i^{(1)} - p_i^{(0)}) \tag{9}$$

where the subscript x refers to the experimentally measured values, the partial derivatives are evaluated as $T_i^{(0)}$, $p_i^{(0)}$, and the identification of dN with $N_{(x)} - N^{(0)}$, dT_i, with $T_i^{(1)} - T_i^{(0)}$, etc., is obvious. Since the values for $T_i^{(0)}$ and $p_i^{(0)}$ were assumed, the only unknowns in the system (9) are $T_i^{(1)}$, $p_i^{(1)}$; they occur in a 2-by-2 linear system, and the solution for them is immediate.

The $T_i^{(1)}$, $p_i^{(1)}$ is a first approximation to the true value T_i, p_i for the unknown zone. However, the eqns (1) and (4) are quite nonlinear in T_i, p_i and eqn (8) assumes a linear relationship. Thus, the process must be repeated using $T_i^{(1)}$, $p_i^{(1)}$ to calculate a new N_1, τ_1, which in turn will lead to a $T_i^{(2)}$, $p_i^{(2)}$. The technique is iterated until the values of N, τ calculated agree with the experimentally measured values within predetermined limits. The converged value is then the T, p ascribed to that zone of the gas.

Hence, for the second zone, since any reasonable guess is satisfactory for the zeroeth approximation, start with $T_2^{(0)} = T_1$ and $p_2^{(0)} = p_1$. The values calculated from eqns (1) and (4) are compared with N_b, τ_b which suggests the $T_2^{(1)}$, $p_2^{(1)}$, and the process is iterated until the convergence criteria are satisfied.

B.   INPUT DATA PREPROCESSING CODE SUMMARY

This code takes inputs from four different sources to produce the input file required to run the AEDC deconvolution code:

1) Manually input zone locations.

2) Standard Plume Flowfield (SPF) model.

3) ES 100 spectrometer.

4) Weighted infrared spectrum filter file (FILTER.dat).

Dr. Jose Sinibaldi, Research Assistant Professor at NPS, Department of Aeronautics and Astronautics, and my co-advisor, created the input data preprocessing code in Microsoft Visual C++. When the "ETDATA.exe" program is executed, an NT shell window opens and asks the user a series of questions regarding the downstream position of the rotation / translation scanning mechanism, the wavelength band of interest, and the primary and secondary combustion species of interest. A single deconvolution code input file (see Appendix B) calculates concentration and temperature data for half the area of the exhaust plume (assumes a symmetric distribution) and for one particular combustion species. When the deconvolution code was created by AEDC, it assumed axisymmetric flow so only half the plume required scanning. The mechanism used for this research scans through both halves of the plume, where the plume is small enough, to create two separate two-dimensional axisymmetric images. This additional information was used to create the nonsymmetric two-dimensional images at each downstream location (explained in the Results section). The following summary of the information contained in the input file explains how the four sources of data are used.

The first line of the input file contain values for the Number of combustion Species the code will consider (NS) and the Index for Species for ITERation (ISITER). Both values are always 2. The deconvolution code will treat the second species of the input file as the primary (to be solved) and the first species as the secondary (considers radiation crosstalk at the particular band). The second line of the input file contains five numbers: NTOT, R0, IDIR, MODEL, ISPF. NTOT is the total number of zones scanned, 10 for this thesis. A "zone" is defined as a concentric ring of the plume, of constant thickness. R0 defines the outer zone radius at 0. IDIR of -1 tells the deconvolution code to solve the "inverse" problem: from emission and transmission, calculate concentration and temperature. MODEL of value 1 describes which species database model to use, SIRRM in this case (explained in Section III.C). ISPF describes how the zones are

9

defined in the plume; outer edge to the centerline for a value of 0. The third line of the input file is simply text: the name of the file. For example, the filename "X05DA045_AEDC.dat" refers to the diagnostic equipment measuring data 0.5 rocket nozzle exit diameters downstream from the edge of the nozzle, at a rotary stage angle 45 degrees from the initial position.

The input file then lists 7 columns of numbers, each 10 values in length for the 10 zones measured at each rotary position. The first column, RR, lists the radial position (cm) of each zone from the center of the plume. The second column, PP, lists static pressure predictions (atm) from the SPF model. The SPF model is the current state-of-the-art rocket exhaust plume flowfield prediction model to which the results of this thesis were compared. The conditions input to the SPF model, matching the rocket test cell configuration for the firings, were 1.2 O/F (oxidizer/fuel ratio), 365 psia chamber pressure and 4.1:1 supersonic expansion ratio (no afterburning). The third column, TT, lists static temperature predictions (K) from the SPF model. The fourth and fifth columns are the secondary and primary combustion species (mole fraction) predictions from SPF. The primary species addressed in this research, visible to the spectrometer through the fiber optic cable in the infrared region (approximate window of 1.2 to 4.8 μm), are H_2O, CH_4, CO_2, and CO. The user selects the wavelength band of interest for the deconvolution code to consider, narrow enough to maximize the intensity contribution of only that primary species. The secondary species is chosen if there is some unavoidable crosstalk intensity in the primary species selected wavelength band. The secondary species available for consideration (included in the SIRRM database) are H_2O, CH_4, CO_2, CO, NO, and OH. The user can also select NONE for a secondary species, if there is no crosstalk from the 6 available species in the primary species wavelength band. Even though the end result of this thesis is the determination of the primary species concentration, the deconvolution code requires a primary species concentration estimate column in the input file.

The sixth and seventh columns of the input file are the radiance and transmission values for each zone, as determined from the ES 100 spectrometer measured radiance ($kW/m^2/sr/\mu m$). The radiance is simply the average spectral values of the plume spectral

10

images measured (see Section II.D.) while the shutter is closed, hiding the blackbody. These average spectral values are then integrated over the wavelength bandwidth specified to produce one radiance value for that zone (kW/m²/sr). The AEDC deconvolution code expects the radiance as (W/cm²/sr), so the values are divided by 10 for proper units. The transmission (τ) for each zone is calculated in eqn (10), using these averaged, integrated radiance values.

$$\tau = \frac{N_{P+BB} - N_P}{N_{BB}} \tag{10}$$

N_{P+BB} = radiance measured from exhaust plume and blackbody (shutter open)

N_P = radiance measured from exhaust plume only (shutter closed)

N_{BB} = radiance measured from blackbody only, prior to test / no exhaust plume

This equation calculates transmission as a fraction (0 – 1) of the energy that transmits from the blackbody through the plume to the fiber tip.

Below the 7 columns, the input file lists a value for NFILT, which is the number of wavelength steps the deconvolution code will consider for that particular primary species within that particular wavelength band. The user specifies the lower wavelength limit and upper limit, so the input file lists all wavelength steps between those limits. Each of the 160 elements of the spectrometer's diode array represents a wavelength step. For example, diode 1 corresponds to the spectrometer's lower range of 1.2 µm, while diode 160 corresponds to the spectrometer's upper limit of 4.8 µm; the remaining 158 diodes represent wavelength steps evenly spaced over the 3.6 µm band in between the limits. All 160 wavelength step values are contained in the file "FILTER.dat," as the first column (variable WFILT). The second column also has 160 values, corresponding to the "weighting fraction" for each wavelength step (variable RFILT). This weighting fraction takes into account the inefficiencies of the collection lens and fiber optic cable. In a perfect scenario, the spectrometer would be small, light, and robust enough to quickly translate and rotate around the exhaust plume, horizontally opposed to the shutter and blackbody. Since this state-of-the-art spectrometer is large, cumbersome, and not designed for rapid movement during operation, fiber optic cable is used to transfer the

11

measured signal instead. However, significantly less intensity is transferred to the spectrometer. The weighting fraction values, established from measuring the blackbody reference source at the selected operating temperature, account for the signal loss due to the fiber at each wavelength step, as if the spectrometer were mounted directly to the translation / rotation mechanism.

All emission values recorded by the ES 100 have the background emission subtracted out. Before each test run, the spectrometer recorded 6000 images of the background (no plume or blackbody). These spectral values are also averaged (see Appendix E).

C. AEDC COMBUSTION SPECIES DECONVOLUTION CODE SUMMARY

This combustion species deconvolution code was originally developed for measurements of the Titan II Aerojet Liquid Rocket Engine (LRE) plume at the Arnold Engineering Development Center (AEDC) in Tennessee. The emission / absorption (E/A) source and receiver combination translated from across the plume to the centerline. Axisymmetric plume flow was assumed. The code relies on the Standard Infrared Radiation Model (SIRRM) to compute emission and transmission given a species concentration and temperature. SIRRM has the ability to take spectral information for many different molecular species at different infrared wavelengths, temperatures, and pressures and compute the overall expected radiance. SIRRM version 3.0 was used to complete this thesis work. The deconvolution code ran on a Silicon Graphics Indigo 2 workstation.[4,5,6,11]
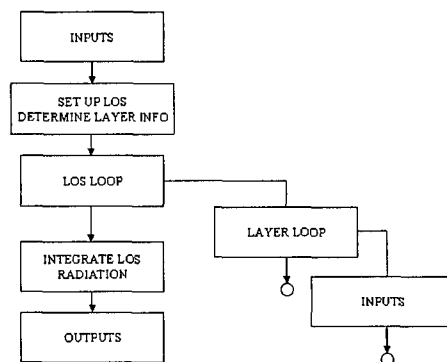
Figure 3.   SIRRM Logic Flow.

12

MAIN Program

INPUT

FILTINT

FINDCL

SETSIR

INVERT

FORWARD

CHOLES

Subroutines

WRTSIR

SYSTEM
(SIRRM)

RDSIR

Writes Output to File
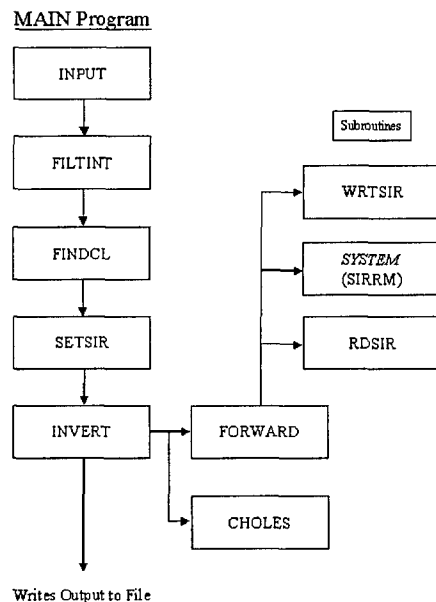
Figure 4.   AEDC Deconvolution Code Structure.

The MAIN program takes the input information of radiance and transmission from the spectrometer, and determines through a numerical program the concentration and temperature of a particular species in each scanned zone.   Information for each targeted molecular species is contained in the SIRRM data files.   Filter information for discrete wavelength increments is created to account for all aspects of the hardware setup, so that any losses or inefficiencies between the actual radiance and transmission at the plume and the information that is provided to the deconvolution code are taken into account.   In this case, it is the transmission characteristics of the fiber optic cable.   The different subroutines are explained below in the order the MAIN program calls them.

The INPUT subroutine reads the input file, which contains information acquired from the plume by the scanning apparatus (radial position, radiance, and transmission for each plume zone) as well as data produced from an equilibrium model (SPF) that predicts conditions for the particular test conditions, such as static pressure, static temperature, and concentration of "crosstalk" species for each plume zone.   This subroutine also reverses the discrete wavelength column information (setting up for FILTINT

13

subroutine), as well as normalizes the filter weighting values for each wavelength (acquired from FILTER.dat).

The FILTINT subroutine converts the wavelength band lines in microns to wavenumber (cm^{-1}), and interpolates each value to the nearest 5 cm^{-1}. This is the increment that the SIRRM code expects to crosscheck with its species data files. The FINDCL subroutine determines the characteristic length array of the plume along the line of sight. The SETSIR subroutine creates an input file for SIRRM that contains the upper and lower filter limit wavenumbers for the particular species band, as well as the wavenumber increment value (5 cm^{-1}).

The INVERT subroutine is the core of the program that runs the SIRRM software. Given the intensity and transmission "target" measurements acquired from the ES 100 at a particular zone (line of sight across the plume at a particular radial position), and the static pressure, temperature, and concentration predicted at that zone by SPF, the SIRRM code guesses a species temperature and concentration. This guess is specific to that specific wavenumber band being analyzed for only that one particular molecular species. SIRRM then calculates a "current" radiance and transmission based on that guess. The INVERT subroutine compares the "target" and "current" values against a convergence criterion. If the criterion is not met, the code steps the concentration and temperature values in the proper direction to improve the convergence. The program iterates a maximum of 20 times to meet the convergence criteria.
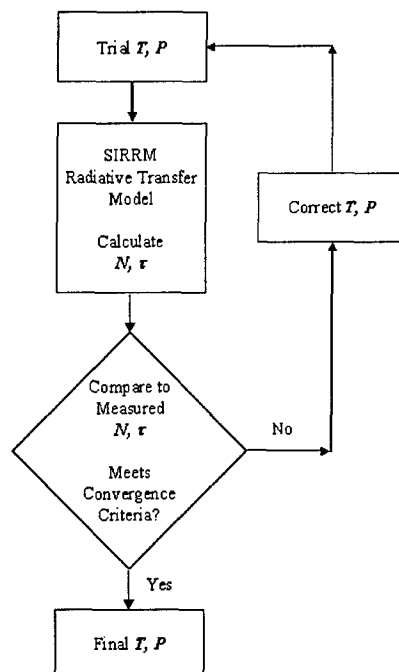
Figure 5. AEDC Deconvolution Code Logic Flow.

The AEDC Combustion Species Deconvolution Fortran code and the SIRRM code are not included as appendices due to distribution limitations. The data is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.). Those interested in obtaining the codes are asked to contact Mr. Daniel Brown at AEDC for the deconvolution code and Mr. Thomas Smith at U.S. Air Force Phillips Laboratory for the SIRRM code. Both individuals and their addresses are listed in the Initial Distribution List at the end of this thesis.

D.   TEST MATRIX

The diagnostic scanning equipment was placed 3.0 exit diameters downstream of the rocket motor exit. The exit diameter is 1.54 cm, placing the E/A imaging line 4.63 cm downstream. Two separate rocket motor oxidizer-to-fuel (O/F) ratios were tested: 1:2 and 1:5. The SPF code was run to simulate a 1:2 O/F condition. After it was determined higher plume temperature was required to boost the intensity signal, a 1.5 O/F condition was tested. The first test at 1:5 O/F ratio began at the SPF predicted edge of the plume,

1.2 cm above rocket exit nozzle centerline, and stepped down 20 zones of 1.2 mm each to the bottom edge of the plume, 1.2 cm below centerline. For the second case, at 1:5 O/F ratio, the translation began at 1.8 cm above centerline, well above the plume boundary at 3.0 exit diameters. By translating 1.8 mm between each E/A measurement, this allowed us to characterize the plume boundary according to the zone that provided the first emission measurement. We accepted the fact the plume would not have 10 zones per radius for this test.

The translation / rotation scanning mechanism testing sequence of events is run by the ESP300 controller. A serial port communications program, written specifically for our testing in Microsoft HyperTerminal, is used to control the ESP300 controller. The program for the 3.0 diameter downstream location, 1.2 O/F ratio test, with command descriptions, is shown below in Table 1. The left hand column is the actual code, which was created as a simple text file, then copied and pasted into the HyperTerminal window for execution. The program for the 1.5 O/F ratio test is virtually identical. If the rotary stage were actuated during a test run, this code would control its motion also.

```
1XX       #CLEAR PROGRAM 1 FROM MEMORY, IF ANY
1EP       #ACTIVATE PROGRAM MODE, ENTER THE FOLLOWING COMMANDS AS PROGRAM 1
BO1H      #SET PORT A FOR OUTPUT
SB3H      #SET PORTS #1 & #2 TO 5.0 VDC, OR TTL HIGH
                   (SHUTTER OPEN, SPECTRALINE SOFTWARE WAITING TO RECORD DATA)
1PA0.0    #MOVE LINEAR STAGE / AXIS 1 TO POSITION 0.0
1WS       #WAIT FOR AXIS #1 TO STOP
SB2H      #SET TTL #1 TO LOW, RECORD TRANSMISSION DATA
WT110     #WAIT 110 MS (ADEQUATE TIME FOR SPECTRALINE TO TAKE 10 SAMPLES)
SB3H      #SET TTL #1 TO HIGH, STOP RECORDING TRANSMISSION DATA
WT390     #WAIT 390 MS (ADEQUATE TIME FOR PC TO CREATE FILE ON HARDDRIVE)
SB1H      #TTL #2 TO LOW (SHUTTER CLOSES)
WT5       #WAIT 5 MS FOR SHUTTER TO CLOSE
SB0H      #SET TTL #1 TO LOW, RECORD EMISSION DATA
WT110     #WAIT 110 MS (ADEQUATE TIME FOR SPECTRALINE TO TAKE 10 SAMPLES)
SB1H      #SET TTL #1 TO HIGH, STOP RECORDING EMISSION DATA
WT390     #WAIT 390 MS (ADEQUATE TIME FOR PC TO CREATE FILE ON HARDDRIVE)
SB3H      #SET TTL #2 TO HIGH, OPEN SHUTTER
WT5       #WAIT 5 MS FOR SHUTTER TO OPEN
1DL       #DEFINE LABEL 1
```

16

```
1PR+1.2          #MOVE AXIS #1 1.2 MM FROM CURRENT POSITION
1WS              #WAIT FOR AXIS #1 TO STOP
SB2H             #SET TTL #1 TO LOW, RECORD TRANSMISSIN DATA
WT110            #WAIT 110 MS (ADEQUATE TIME FOR SPECTRALINE TO TAKE 10 SAMPLES)
SB3H             #SET TTL #1 TO HIGH, STOP RECORDING TRANSMISSION DATA
WT390            #WAIT 390 MS (ADEQUATE TIME FOR PC TO CREATE FILE ON HARDDRIVE)
SB1H             #TTL #2 TO LOW (SHUTTER CLOSES)
WT5              #WAIT 5 MS FOR SHUTTER TO CLOSE
SB0H             #SET TTL #1 TO LOW, RECORD EMISSION DATA
WT110            #WAIT 110 MS (ADEQUATE TIME FOR SPECTRALINE TO TAKE 10 SAMPLES)
SB1H             #SET TTL #1 TO HIGH, STOP RECORDING EMISSION DATA
WT390            #WAIT 390 MS (ADEQUATE TIME FOR PC TO CREATE FILE ON HARDDRIVE)
SB3H             #SET TTL #2 TO HIGH, OPEN SHUTTER
WT5              #WAIT 5 MS FOR SHUTTER TO OPEN
1JL18            #JUMP TO LABEL 1 18 TIMES
WT500            #WAIT 500 MS TO ENSURE ALL MOVEMENT HAS STOPPED
QP               #QUIT PROGRAM
```

Table 1.   ESP300 Controller Diagnostic Testing Code.

Spectraline created a custom designed GUI executable program, TriggerData.exe, for this research. The program writes to the harddrive and creates a file every time a TTL low signal is received from the ESP300 controller. 40 Spectraline *.scn files were created for each test run; scanning through the entire plume, there were 20 total zones, with an emission and absorption measurement recorded for each zone. Each *.scn file includes 10 spectrometer images, created at 390 Hz (0.00256 seconds for one image). After the run, each file is individually opened in Spectraline and a "statistics" text file was created, which averages the radiance at each pixel or wavelength over the 10 images.

THIS PAGE INTENTIONALLY LEFT BLANK

III.   EXPERIMENTAL SETUP

A.   COMMERCIAL HARDWARE

1.   Spectraline ES 100 Imaging Spectrometer

The ES 100 is a fast imaging spectrometer (see Figure 6) that provides spectral radiation intensity in the mid infrared region (MIR), 1.2 to 4.8 μm (see Figure 7). In this region, the combustion species spectral signatures emanate from strong fundamental bands that are widely spaced apart compared to the near infrared region (NIR, 0.7 to 2.5 μm), providing an inherently high signal to noise ratio. The mid infrared region is not affected by environmental influences such as sunlight or room light. The lengthy calibration process therefore required for NIR instruments (6 months to a year) is avoided.[7]

The ES 100 infrared spectrometer is currently the fastest and smallest of its kind. The ES 100 measures the MIR spectrum at a repetition frequency of 390 Hz, making it an ideal instrument to characterize transient phenomena such as unsteady combustion. In comparison, the fastest FTIR (Fourier Transform Infrared) spectrometers collect an interferogram of a sample signal using an interferometer, then perform a fourier transform on the interferogram to obtain the spectrum at only 1 Hz. The ES 100 is housed in a 9.5 x 10.5 x 4.5 inch cast aluminum enclosure, and contains no gratings or moving mirrors. Instead it uses reflective optics and prisms for spectral dispersion. The spectral intensity entrance port is a slit 6 mm high and 0.5 mm wide. The energy passes through two vibrating ribbons (operating at 390 Hz) that act as a chopper (produce an alternating signal since the diode array responds to alternating current), through two reflective parabolic mirrors, and is then dispersed by a series of two calcium fluoride prisms into its spectral components. This dispersed radiation is imaged onto a 160-element lead selenide array. This detector, unlike most spectrometers, does not require cryogenic cooling; it uses a thermoelectric cooler which has a lifetime 100 times longer than cryogenic cooling. The use of fully reflective components for imaging, and prism elements for dispersion, provides the highest possible throughput for a MIR spectrometer.[8,9,10] A penny is displayed next to hardware in photographs for relative size comparison.

19
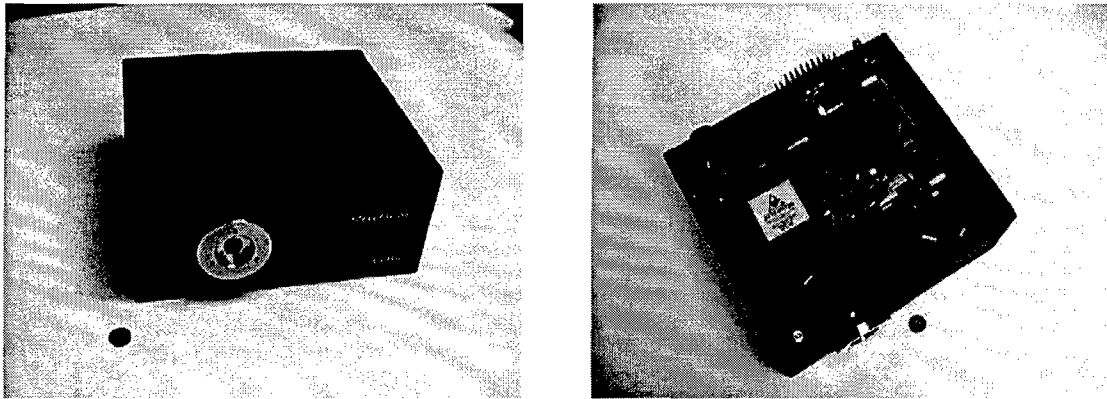
Figure 6.   Spectraline ES 100 Imaging Spectrometer.
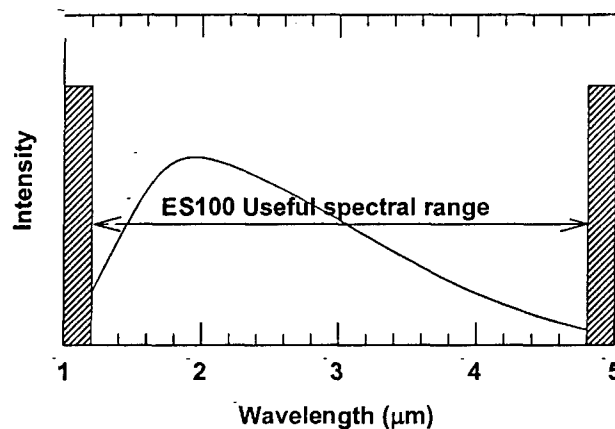
Figure 7.   ES 100 Spectrometer Range.

2.   Newport RV160CC Rotary Stage

The RV160CC Rotary Stage (see Figure 8) was chosen for several reasons. To satisfy the testing requirement to gather as much data as possible during each rocket engine firing, the stage rotates up to 80 degrees per second. The stage can support a maximum of 675 N (152 lbs) at the mounting plane. However, operating at these maximum speeds and loads requires 2 amps of current. Thus, a custom-built 2-amp computer card is required for the Newport ESP300 Motion Controller/Driver. The controller uses readily available 1-amp cards. Due to funding constraints and the long lead-time required for the vendor to produce the 2-amp card, only a 1-amp card was used for this thesis. Therefore, the rotary stage was limited to only 1 degree per second, allowing rotation of the mechanism only between engine firings. The controller actuates

20

the motor to a resolution of 0.00075 degrees. This precise rotation is ensured by a precision ground, hardened worm and gear drive, that provide smooth, continuous motion.
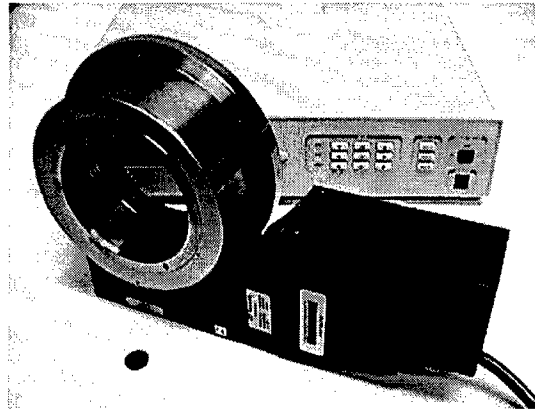
Figure 8.   Newport RV160CC Rotary Stage and ESP300 Motion Controller/Driver.

3.   Newport 850G-HS Linear Actuator

In order to rapidly translate the fiber optic cable (receiver) and the shutter / blackbody arm (emitter) through the plume diameter, the 850G-HS (High Speed) Linear Actuator (see Figure 9) was selected. The closed-loop DC motor drives the tip arm up to 6 mm/sec, up to a maximum travel of 50 mm. The tip arm moves the linear stage, to which the diagnostic hardware attaches. The minimal incremental motion is ≤ 1 μm at accuracy of <0.1% of the travel, a specification necessary to precisely position the diagnostic hardware in plume zones that range from 0.9 – 1.5 mm width. The motor can drive up to 2.3 kg at high speed, and uses the same Newport ESP300 Motion Controller/Driver as the rotary stage. The actuator only requires the 1-amp card in the controller to operate at maximum limits.
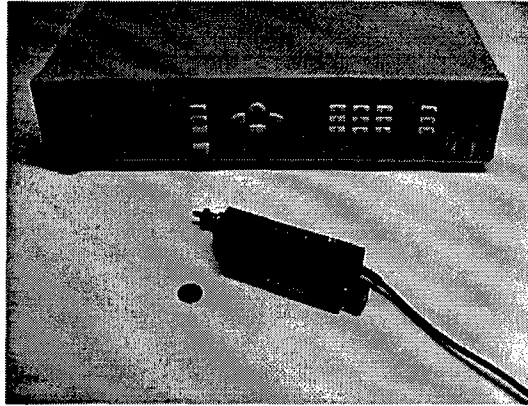
Figure 9.   Newport 850G-HS Linear Actuator and ESP300 Motion Controller/Driver.

4.   Newport 436 Linear Stage

The linear stage (see Figure 10), moved by the linear actuator, connects the translation and diagnostic hardware to the rotation hardware. The crossed-roller bearing design running on the steel V-ways allows for precision movement (angular deviation of <150 µrad) and high load capacity (125 lbs / 556 N center load) of the parallel aluminum plates. The stage maximum travel is 2 in / 50 mm. This hardware was primarily chosen for its compatibility with the Newport 850G-HS Linear Actuator.
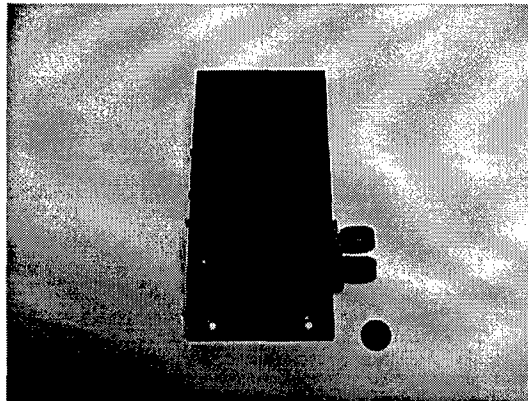
Figure 10.   Newport 436 Linear Stage.

5. Oriel 76992 High Speed Shutter

The high-speed shutter (see Figure 11) can cycle between fully open and fully closed at rates up to 200 Hz. The 6 mm circular aperture has two overlapping blades that are coated with aluminum silicon oxide (AlSiO) for high reflectance on the input (blackbody) side. This decreases the radiation absorbed on the input side and the overlapping prevents light leakage. The Oriel 76995 Electronic Shutter Controller receives TTL pulses to actuate the shutter.
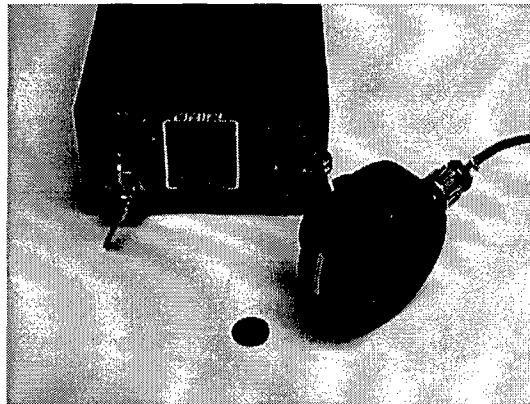
Figure 11. Oriel 76992 High Speed Shutter and Electronic Shutter Controller.

6. Boston Electronics IR-508 Blackbody

The IR-508 blackbody (see Figure 12) provides infrared radiation as an ideal blackbody emitter. The output energy from the 0.25-inch cavity, generated by a ceramic-sealed heater coil, closely follows the theoretical maximum energy curve described by Planck's equation. The low weight (2.5 lbs) and low power requirement (2.0 amp max at 110-120 VAC) make it ideal to mount behind the high-speed shutter on the translation / rotation mechanism. The blackbody has a temperature range of 50 degrees C to 1050 degrees C, with an accuracy of +/- 1.5 degrees C. This hardware uses the Boston Electronics IR-201 Digital Temperature Controller.
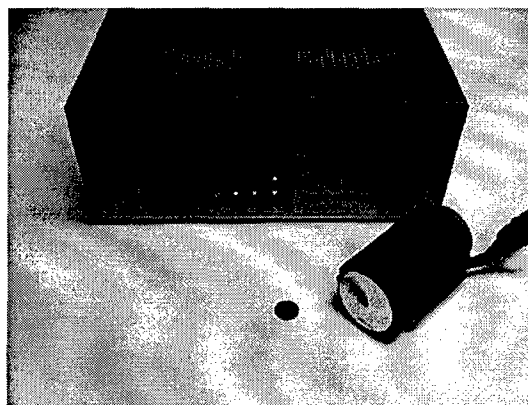
Figure 12.   Boston Electronics IR-508 Blackbody and IR-201 Digital Temp. Controller.

7.   Thor Labs 0.28 NA Fluoride Glass Fiber Optic Cable

The Thor Labs fiber (see Figure 13) transmits mid infrared radiation (MIR) from approximately 0.5 μm to 4.0 μm, transmitting nearly the full range of viewable energy to the ES 100 spectrometer (views 1.2 μm to 4.8 μm).   The 200 μm cladding diameter covers the 160 μm core, with standard SMA connectors attached at both ends at special request.   The 5 m fiber optic cable has a minimum bend radius of 20 mm, allowing significant flexibility between the spectrometer and the mechanism to transmit usable radiation as the stage rotates and translates during testing.   The numerical aperture (NA) is 0.28.
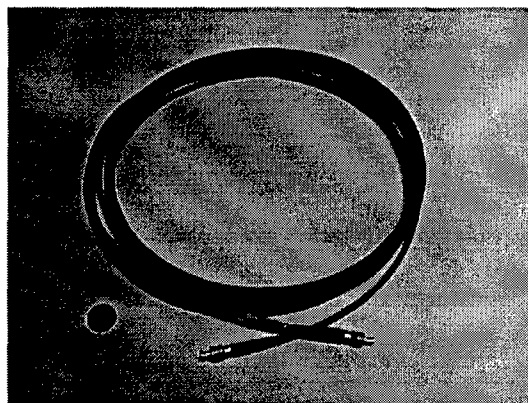
Figure 13.   Thor Labs 0.28 NA Fluoride Glass Fiber Optic Cable.

24

Originally, an Oriel 0.2 NA fluoride glass 600 µm core fiber (see Figure 14) was chosen for this research. Although much less flexible (minimum bend radius of 10 cm), the cable transmitted a larger band of infrared radiation, from approximately 0.5 µm to 4.5 µm. Unfortunately, the metal cladding of the fiber tip separated from the SMA connector at one end (defective manufacturing), breaking the fluoride glass core. Since there were no replacement fibers available, the Thor Labs fiber was used as the primary backup.
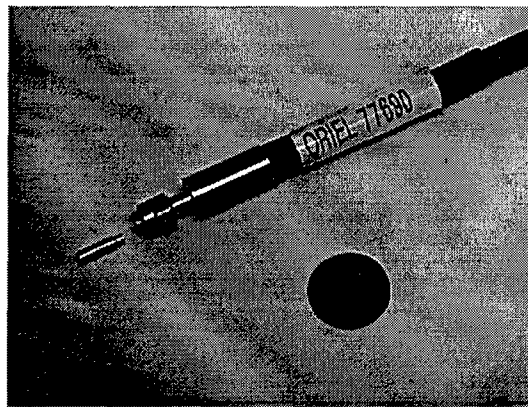
Figure 14.   Broken Oriel Fluoride Glass Fiber Tip.

8.   Edmund Industrial Optics 3.18 mm Diameter Sapphire Ball

Where the end of the fiber optic cable measures the intensity from the rocket exhaust plume and blackbody, a ball lens (see Figure 15) is required to focus the collimated intensity into the fiber at the proper angle. The Fiber Optic Cable Mount (see Section III.B.4.) holds the ball lens against the fiber tip. Sapphire is an excellent transmitter of the MIR region, as well as an extremely hard material. The required ball diameter calculation is explained in Section III.B.4.
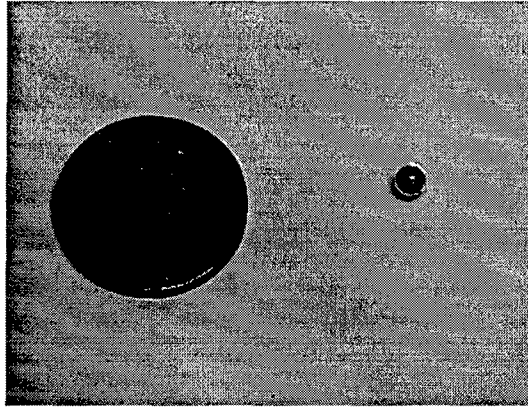
Figure 15.   Edmund Industrial Optics 3.18 mm Diameter Sapphire Ball.

9.   Melles Griot 10.0 mm Diameter Plano-Convex Sapphire Lens

Two lenses (see Figure 16) of identical specification are necessary to transmit enough energy from the exhaust plume to the spectrometer diode array. Where the end of the fiber optic cable interfaces to the spectrometer, the first lens is required to collimate the light leaving the fiber. The Fiber Optic Cable Collimator (see Section III.B.5.) holds the sapphire lens at the paraxial focal length of 10 mm from the tip of the fiber. A 10 mm focal length lens with 10 mm diameter was readily available from the vendor. The second lens collimates energy from the blackbody across the 19.5 cm gap to the Fiber Optic Cable Collimator.
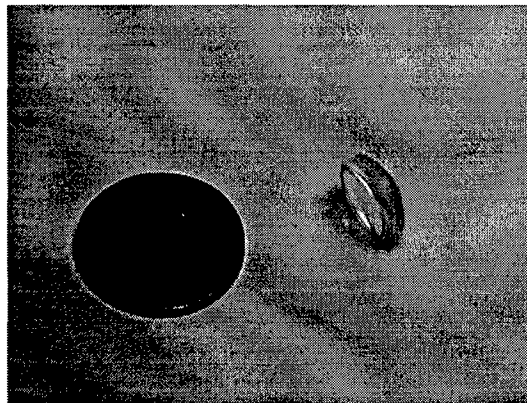
Figure 16.   Melles Griot 10.0 mm Diameter Plano-Convex Sapphire Lens.

26

B. CUSTOM HARDWARE

1. O-Bracket

This ring shaped ¼-inch aluminum bracket (see Figure 17) holds the emission and transmission diagnostic equipment that looks at and through the exhaust plume. The bracket attaches to the linear stage, which translates the diagnostic equipment through the entire cross-section of the plume. The Z-Bracket (see Section III.B.2.), holding the shutter and blackbody mounts on the left, and the Fiber Optic Cable Mount attach on the right side of the plate. Additional ¼-20 thru-holes are drilled to allow for multiple alignment positions. The bottom end of the bracket is cut to 1/8-inch thickness to allow for maximum rotation clearance.
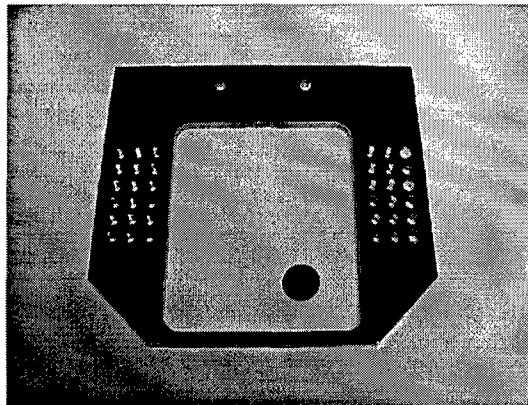
Figure 17.   O-Bracket.

2. Z-Bracket

The Z-bracket (see Figure 18), also ¼-inch thick aluminum, mounts to the O-Bracket. It holds the shutter and the blackbody, so that the centerline of the shutter opening is directly opposed to the fiber optic cable centerline, mounted on the opposite side of the O-Bracket. Additional ¼-20 thru-holes were drilled at the end of the bracket to allow flexible mounting positions for the blackbody. The screws on the bottom side of the Z-bracket for the mounting the shutter and the blackbody were countersunk flush so the mechanism could rotate and not impinge on the rotary stage housing. A shim was added so that the shutter position could not rotate along the axis of the single mounting screw.

27
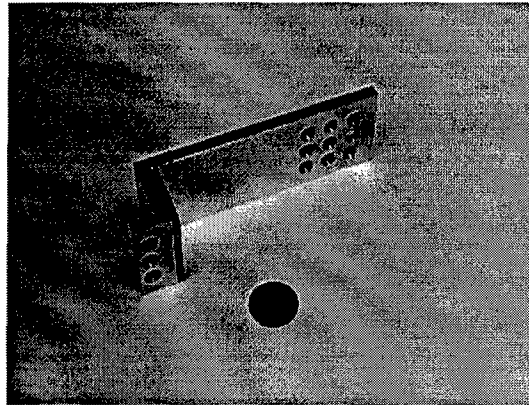
Figure 18.   Z-Bracket.

3.   Linear Stage Lock Mount

The tip arm of the linear actuator pushes against only spring tension internal to the linear stage. With this configuration, the linear stage sliding plate that holds the O-Bracket, and all hardware attached to it, can slide freely away from the tip arm if the spring force is overcome. A copper lock mount piece (see Figure 19) was designed to fasten the sliding plate to the tip arm; the sliding plate moves only when driven by the linear actuator tip arm. One end of the lock mount is threaded and attaches to the sliding plate through a cylindrical mount with a washer and nut. The other end is a cylindrical opening that holds 3/8-inch of the tip arm, secured with a 6-32 screw from the side.
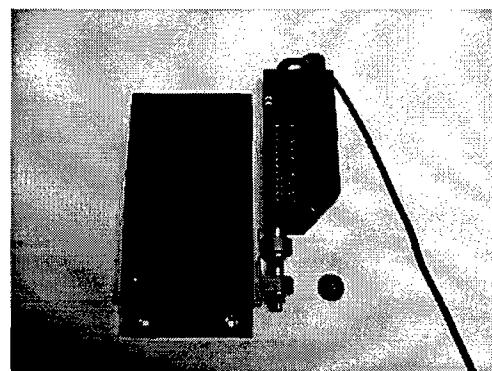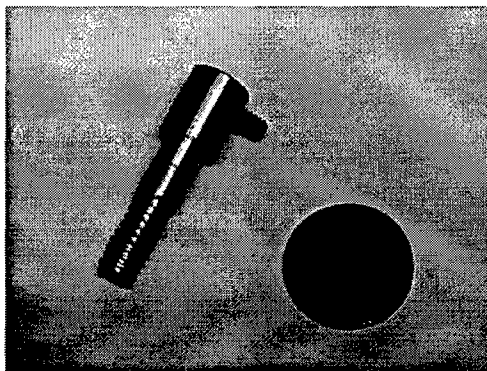
Figure 19.   Linear Stage Lock Mount.

28

4. Fiber Optic Cable Mount

One end of the fiber optic cable mounts to the O-Bracket where it views the emission and transmission spectra of the rocket exhaust plume. The energy collected by the fiber should only emanate from the exhaust plume and the blackbody; therefore a narrow field of view is required. A rectangular aluminum bracket (see Figure 20) has two ¼-20 thru holes for mounting to the O-Bracket. A 2 mm field of view hole allows the plume / blackbody emission into the mount, where the available light then travels through a 1 mm aperture tube. The light then passes through a 3.18 mm sapphire ball, where it is focused onto the tip of the 0.28 NA fiber (See Figure 21). The sapphire ball is held tight between the tip of the fiber, which screws into the other end of the mount through an SMA connector, and the 1 mm aperture tube. The SMA connector internal open diameter was slightly increased along 2/3 of its length provide clearance for the sapphire ball. The sapphire ball diameter of 3.18 mm (commercially available) was calculated from the following equation.

$$NA > \frac{2d(n-1)}{nD} \qquad (11)$$

NA = numerical aperture of the fiber (0.28)

d = diameter of column of energy entering the ball / tube aperture (1 mm)

n = index of sapphire (1.77)

D = ball diameter (3.18 mm)

The 2 mm aperture allows more light into the cable mount. However, focusing that light into the 0.28 NA fiber would require a sapphire ball twice the diameter (6.36 mm). Such a ball would be too large for the SMA connector, requiring custom adapters at the end of the fiber optic cable. For simplicity and spatial resolution, the 1 mm aperture design was pursued.
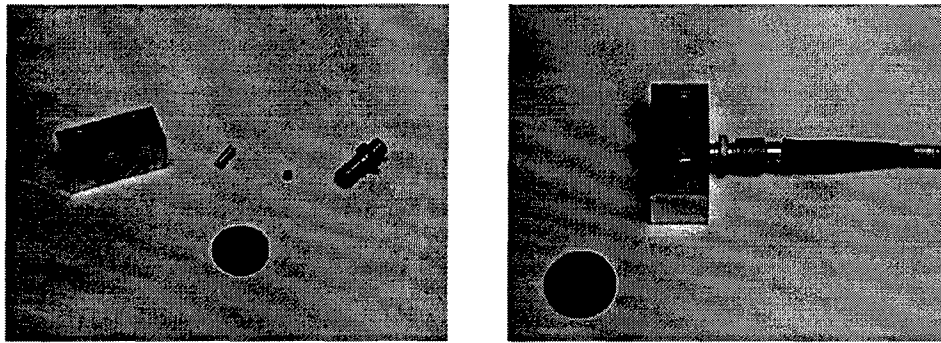
Figure 20.   Fiber Optic Cable Mount.
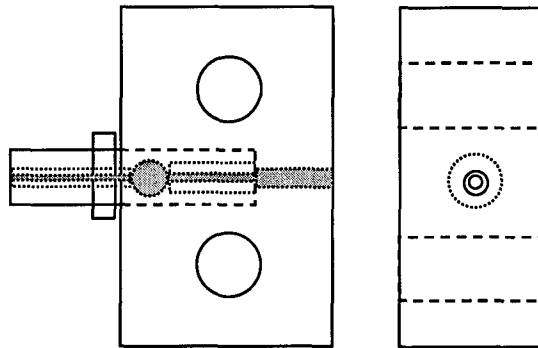
Figure 21.   Fiber Optic Cable Mount Schematic.

5.   Fiber Optic Cable Collimator

The light that passes through the fiber must be collimated at the interface to the spectrometer. In basic mode, the spectrometer is placed near the infrared phenomenon to be measured. The energy passes through a window to the 6 mm entrance slit recessed approximately 25 mm within the spectrometer box. A simple fiber optic cable mounting plate, produced by for this research by Spectraline, attaches the SMA connector end of the fiber to the spectrometer box (see Figure 22), but the fiber tip is still 25 mm from the entrance slit. Any light leaving the fiber tip immediately disperses at an acceptance cone angle dictated by the fiber's numerical aperture (NA). The 10 mm diameter plano-convex sapphire lens, held in place by a 10 mm diameter spring against the mounting plate, catches the expanding light leaving the fiber at a focal length of 10 mm and collimates it across the remaining distance to the entrance slit (see Figure 23). An 0.28

NA fiber has an acceptance cone half angle (a) of approximately 16 degrees, following the equation:

$$NA = \sin(a) \tag{12}$$

At a distance or paraxial focal length (FL) of 10 mm (x), the light leaving the fiber tip will spread to a radius of 2.86 mm (y) from the centerline.

$$\tan(a) = \frac{y}{x} \tag{13}$$

This translates into a 5.72 mm diameter beam ($2a$) of collimated light projected onto the 6 mm entrance slit to the spectrometer. The spectrometer requires at least 4.6 mm of the 6 mm slit be radiated for sufficient detection at the diode array.
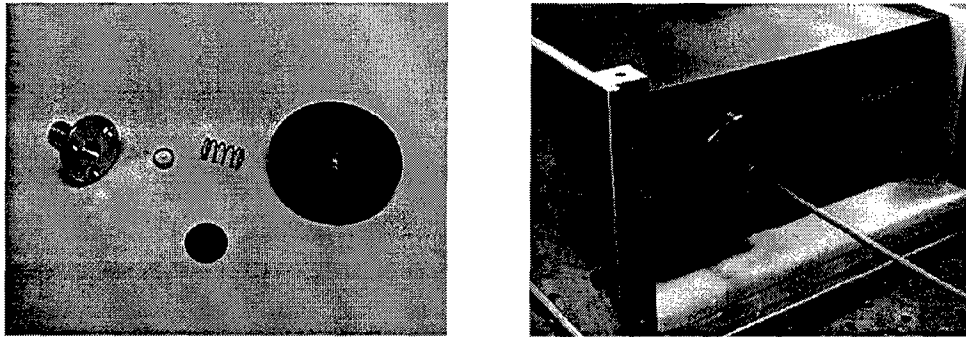
Figure 22.   Fiber Optic Cable Collimator and Mounting Plate.
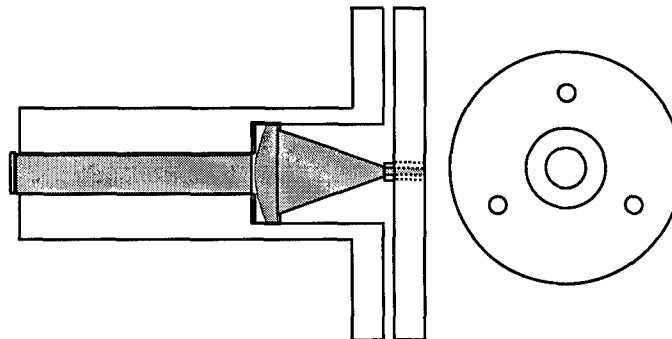
Figure 23.   Fiber Optic Cable Collimator and Mounting Plate Schematic.

31

6. Blackbody Collimator

In order to focus more light intensity across the 19.5 cm gap from the blackbody port opening to the Fiber Optic Cable Mount, a 10 mm diameter sapphire plano-convex lens is secured along the imaging centerline on the back opening of the shutter. A 2-piece aluminum lens holder positions the lens in the shutter with a set screw (see Figure 24). The 10 mm focal length lens was used to match the distance from the lens to the blackbody source.
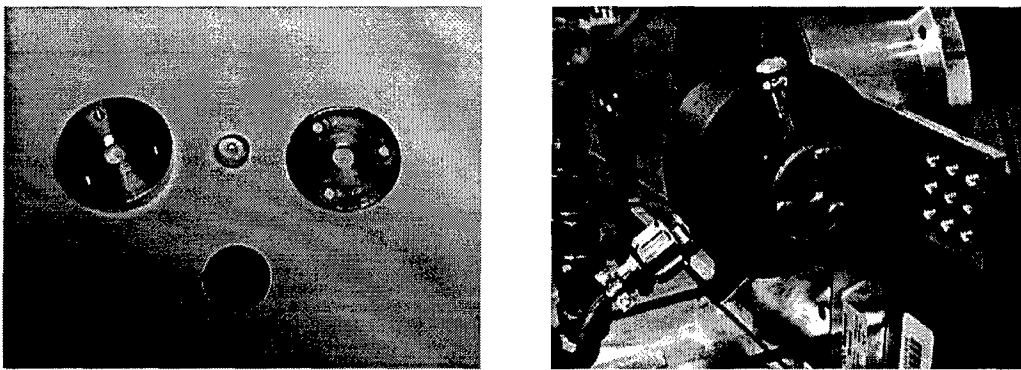
Figure 24.   Blackbody Collimator.

7. Mechanism Support Plates

These two aluminum plates (see Figure 25) interface the translation / rotation scanning mechanism to the rocket motor test stand. The "forward" plate is designed to allow the diagnostic equipment as close the rocket exit nozzle as hardware clearances will allow, approximately 3.7 cm (see Figure 26). The diagnostic equipment mounted to the O-Bracket faces forward towards the motor on this plate. An L-bracket secures the rotary stage and the mechanism to the plate where ¼-20 bolt hole slots are drilled, each 4 inches long, so that the entire plate can be secured for measurements at different downstream locations. Although plume data near the motor exit can be acquired with this configuration, the mechanism can only rotate approximately 90 degrees because the Z-Bracket or O-Bracket will impact the test stand (see Figure 21). With the "reverse" plate configuration, the diagnostic equipment and O-Bracket face away from the motor, allowing the hardware to clear the end of the test stand. This configuration, however, can

32

measure the plume at a minimum downstream location of approximately 12 cm due to clearance issues. The reverse plate also has 4 in. long bolt hole slots for flexible downstream measurement locations, and uses the same L-bracket. Both plates are ½-inch thick, to channel the exhaust plume through the center opening of the rotary stage.
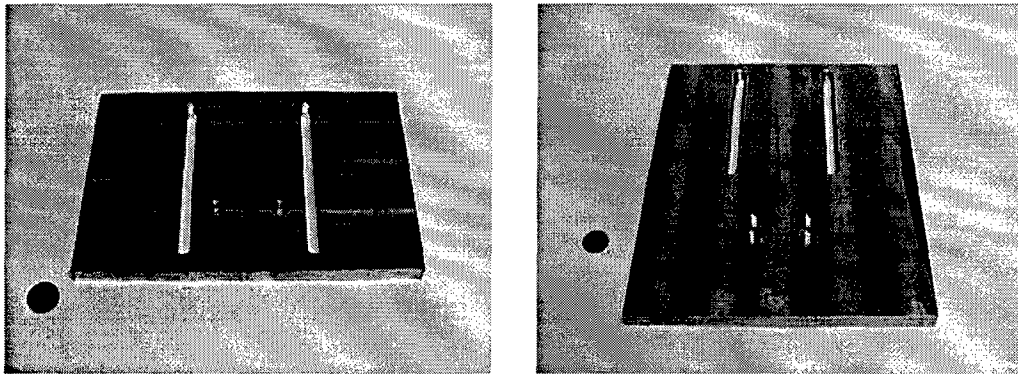
Figure 25. Forward and Reverse Mechanism Support Plates.
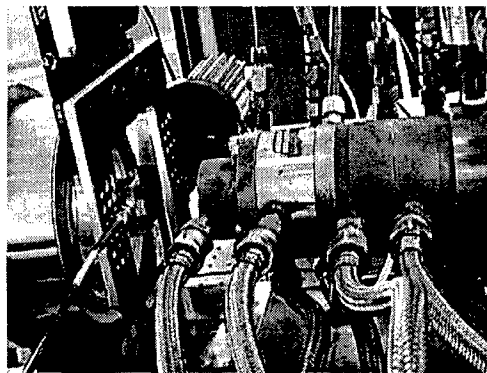
Figure 26. Forward Plate Configuration.

8. Spectrometer Explosion Proof Box

The large cost of the Spectraline ES 100 spectrometer ($50,000), operated within the test cell so the 5 m fiber optic cable could reach the exhaust plume, dictated extra protection for the unit in the event of a rocket engine firing anomaly. The ES 100 was placed in an aluminum box assembled from ½-inch plate on all 6 sides (see Figure 27). The edges and corners of the box are held with ¼-20 bolts attached to internal square columns. Approximately 1 inch of clearance exists for the spectrometer on all sides.

33

With the exception of the back wall, where holes are drilled for the power, data, and nitrogen purge connections, the other 5 side spaces are filled with foam inserts for shock protection. A 2 cm hole was drilled in the front panel for the fiber optic cable interface. The box was held in place on the test stand with ¼-20 bolts.
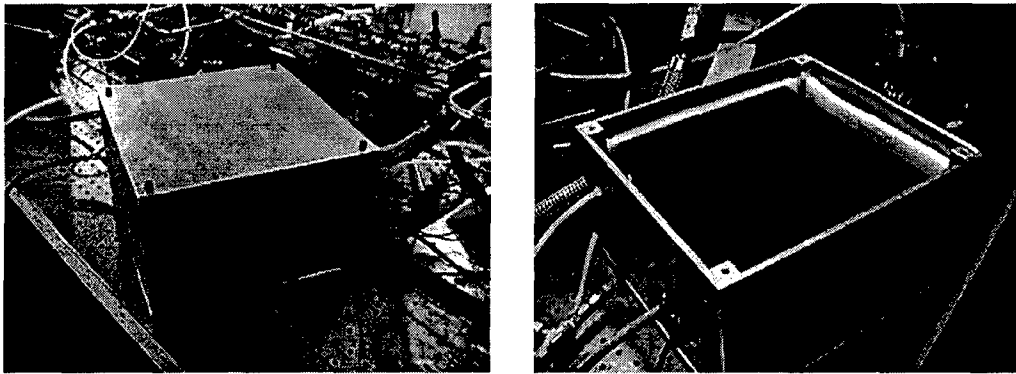
Figure 27.   Spectrometer Explosion Proof Box.

C.   TRANSLATION / ROTATION SCANNING MECHANISM

The entire scanning mechanism is detailed in the pictures below (see Figure 28). The distance between the blackbody port opening and the Fiber Optic Cable Collimator is approximately 19.5 cm. The mechanism can view up to a 5 cm diameter exhaust plume; for a larger plume, heating of the diagnostic equipment and rotary stage seals become a concern. The total cost to produce the apparatus, including the labor involved to build the custom hardware, is approximately $70,000 (ES 100 spectrometer is $50,000 of that total). The integration and total system operation of this hardware and software is depicted in Figure 29.
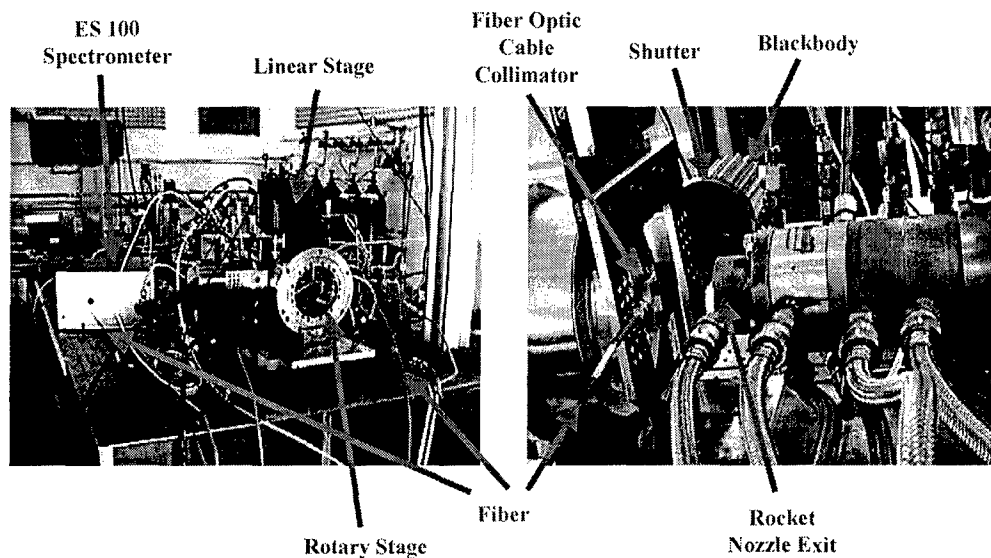
Figure 28.   Translation / Rotation Scanning Mechanism.
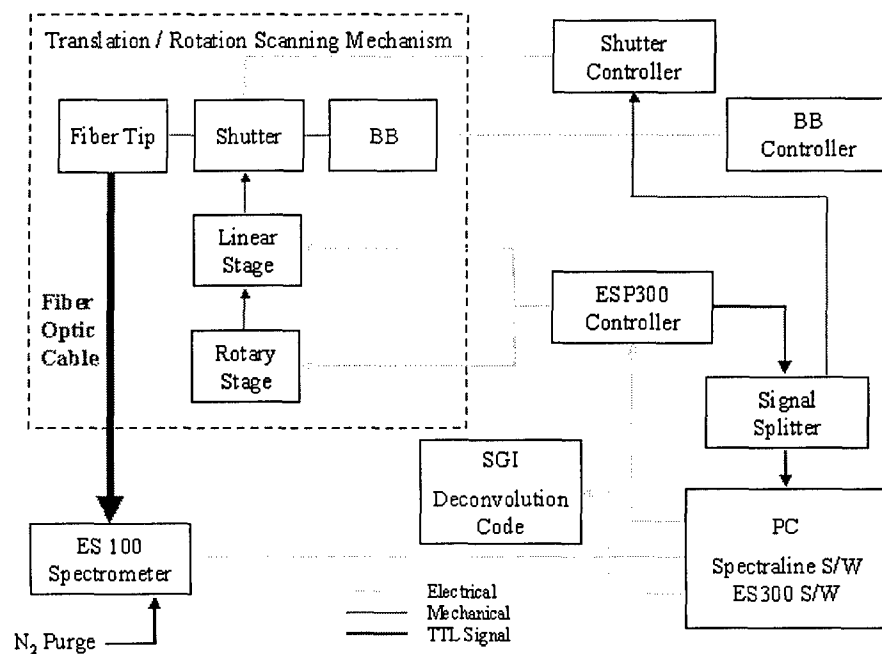
Figure 29.   Scanning Mechanism Total System Operation.

D.   TEST CELL / ROCKET MOTOR

Figure 30 shows the gaseous oxygen / liquid kerosene (O_2/RP-1) rocket motor test cell #3 used to generate the plume exhaust. Four type-K high-pressure oxygen bottles,

connected through a manifold, provided the gaseous oxygen for the rocket engine. A single type-K hydrogen bottle provided the hydrogen for the high temperature oxygen / hydrogen ignition source. Two type-K nitrogen bottles provided valve actuation pressure, fuel pressure, and purge gas. Two 1.8-liter fuel tanks provided the kerosene to the injector. Pressure regulators set the fuel and oxygen flow rates, as well as the nitrogen and film cooling pressures. Five 30-second test runs could typically be performed without replenishing the main fuel tank. Depending on the level of fuel film cooling, between 20 and 50 test runs could be performed before refueling that tank.
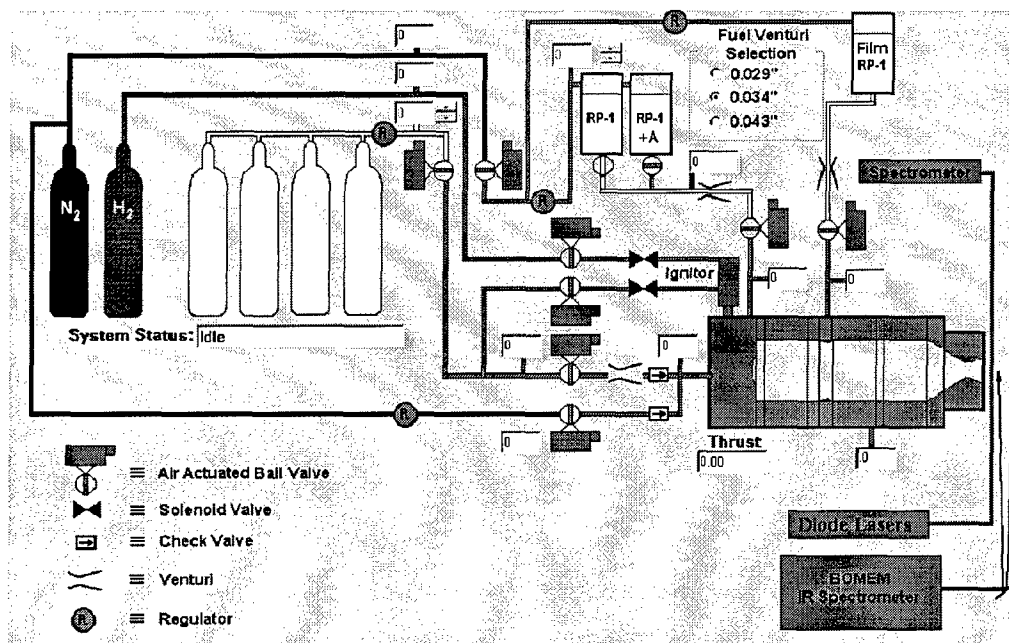
Figure 30. Test Cell #3 Schematic.[2]

Figure 31 shows the rocket engine in the test cell #3 and an exploded view of the motor consisting of the injector head, two water-cooled segments, one heat transfer gauge ring, a pressure tap ring, a transpiration ring, and the converging / diverging Delaval nozzle section. For the limited few tests conducted for this research, the transpiration ring was not used, and was replaced by a third water-cooled segment. All hardware is stainless steel. The nozzle has a 0.3 inch throat diameter, a 4.1:1 exit area to throat area ratio, and an exit angle of 15 degrees, providing a thrust of approximately 35 lbs force.

The injector head has a modular insert so that the engine can be configured for 2, 3, 4, or 5-hole injector depending on the desired fuel flow rate and mixture ratio. The 2-hole injector was used for this research. The fuel injector ports are uniformly spaced in a circular pattern around the oxygen jet inlet. The fuel from each injector impinges along the central axis on this center oxygen streamline jet. Nominal operating pressure for the combustion chamber was 350 psig.
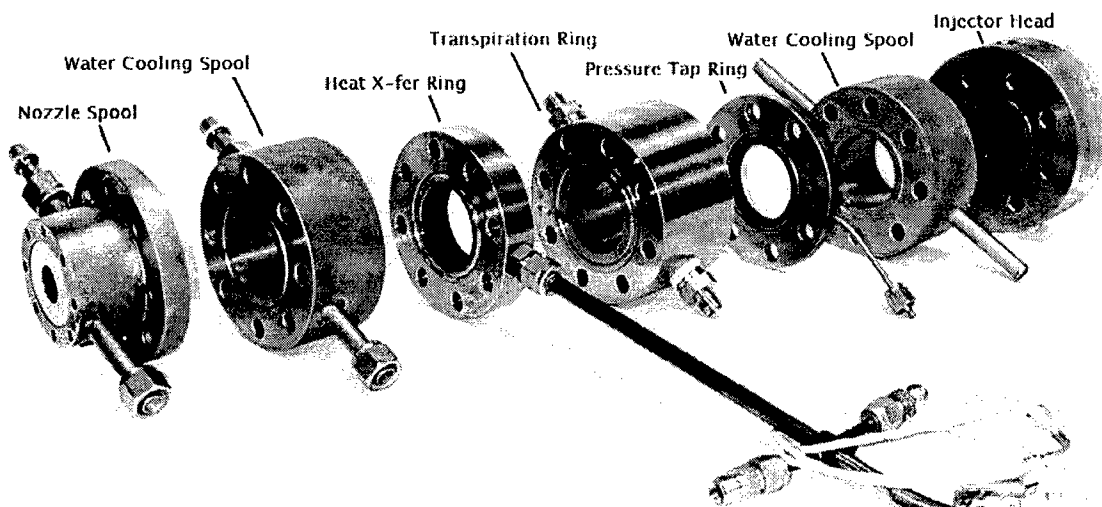
Figure 31.   O₂/RP-1 Rocket Motor and Exploded View (with optional Transpiration Ring)[2].

E.   **SEQUENCE OF TESTING**

    1.   **Pre-Test Checks and Alignment**

For the rocket motor test cell:

1) Press RED "EMERGENCY STOP" button in.

2) Refill kerosene supply tank (ensure no pressure in tank, vent as necessary).

3) Turn on "Xenia" computer, Stanford amplifiers, and power strip.

4) Execute "TEST CELL#3 Control Code" from desktop.

5) Turn on yellow flashing alley lights.

6) Open all gas bottles (RP, pressurized with N_2; H_2/O2 for Torch; Main O_2; Actuator N_2) and set hydrogen pressure on regulator.

7) Verify fuel selection ball valves (YELLOW handles) are in correct position.

8) Verify all igniter and solenoid control lines are connected.

9) Open fuel isolation ball valve (green handle).

10) Evacuate all personnel from test cells.

11) Reset "EMERGENCY STOP" button.

12) In PC (Personal Computer) Control Code "Manual Control", open "N_2 Fuel Pressure" and "O_2 Tank Supply".

13) Set RP and O_2 dome pressures on wall hand loader.

14) In PC Control Code "Manual Control", close "N_2 Fuel Pressure" and "O_2 Tank Supply".

15) Verify correct fuel venturi and run conditions selected in Control Code and modify experiment name.

16) Check for golfers on nearby course holes. When clear, start VCR recording, start water cooling, and sound siren.

17) Press "Start Run" (initiates torch ignition and commencement of run).

For the scanning diagnostic system (accomplished in parallel):

1) Verify nitrogen purge is set to low flow rate for ES 100 spectrometer.

2) Set linear stage to position at top edge of plume, as calculated by SPF model for that particular downstream location. From the PC ESP300 Hyperterminal window, define this position as home (0).

3) Verfiy blackbody is stable at 1050 deg C (max temperature).

4) Verify shutter is closed. The ESP300 controller code will open the shutter with a TTL pulse for the first transmission measurement at the first / outermost zone.

38

5) Verify ESP300 controller software is on for linear stage / axis 1 (view front panel display). If the rotary stage were programmed to move during the firing, ensure the software is on for the rotary stage / axis 2.

6) On the PC, activate the Spectraline "TriggerData" software, and specify the filename and directory to which the *.scn files will be saved.

7) On the PC, activate the HyperTerminal window, and prepare to execute the ESP300 scanning program approximately ½ second after the rocket firing commences (allows plume to reach steady state conditions).

2.   Test Sequence of Events

As the rocket engine fires, the ESP300 controller immediately commands the shutter to open (TTL pulse) and the Spectraline software to "snap" 10 images across the ES 100 signal spectrum, and save those images to a file, called "XXX_0.scn". Another TTL pulse commands the shutter to close, and another 10 images are snapped and written to file "XXX_1.scn". The ESP300 controller now commands the linear actuator to translate a predefined distance to the top of the next zone, and repeat the process for a total of 20 zones (10 zones above and below the plume centerline). The PC software used for this research could only write approximately 2 files per second due to harddrive controller latency issues, limiting the speed of translation between scans.

3.   Post-Test Checks

For the rocket motor test cell:

1) Press "EMERGENCY STOP" button.

2) Verify all valves are closed on Manual Control Form.

3) Turn off warning lights and siren.

4) Exit Control Code.

5) Stop VCR recording.

6) Turn off all gas bottles except N_2 purge.

7) Close fuel isolation ball valve (green handle).

8) Turn off cooling water.

9) Turn off yellow flashing alley lights.

For the scanning diagnostic system:

1) Check the translation / rotation scanning mechanism for damage.

2) Wipe excess water and/or soot off exposed surfaces of the mechanism.

3) From the PC HyperTerminal window, command the ESP300 to return the linear stage to the home (0) position.

IV. EXPERIMENTAL RESULTS

This section discusses two test runs, both tests operated at the desired O/F mixture ratio of 1.5, and the measurements were taken 3 exit diameters downstream of the nozzle exit (approximately 4.63 cm.) "Run 1" scanned the plume from 12 mm above the centerline to 12 mm below, at 1.2 mm intervals thus measuring 20 zones across the plume. "Run 2" scanned the plume from 18 mm above to 18 mm below, stepping 1.8 mm for each zone, in order to determine where the plume boundary began. Before each day of testing, a new weighted infrared spectrum filter was created by imaging the blackbody through the fiber. The blackbody image without the fiber was obtained only during the first day of testing to minimize the uncertainties introduced during assembling and disassembling the fiber optic system. Creating different filters accounted for atmospheric, fiber optic cable, and spectrometer variations between testing periods. Since Run 1 and 2 were accomplished on the same day within the same hour, the atmospheric conditions remained relatively constant and the diagnostic hardware was not disturbed, only one weighted filter was required. Dividing the ES 100 spectrometer spectral radiation reference measurement (obtained without the fiber optic cable) of the blackbody 19.5 cm away (scanning mechanism distance) by the Fiber Optic Cable Collimator spectral radiation measurement from the same blackbody 19.5 cm away produces the weighted values at each wavelength interval. Figure 31 shows the Spectraline plots (wavelength vs. intensity) of the blackbody without the fiber (top) and with the fiber (bottom) taken before testing. Notice the y-axis maximum is 10 for the top plot and 1 for the bottom plot, and that the zero reference is the bottom of the y-axis for the top plot and approximately 2/3 up the y-axis for the bottom plot. Figure 32 depicts the resultant filter for Runs 1 and 2.
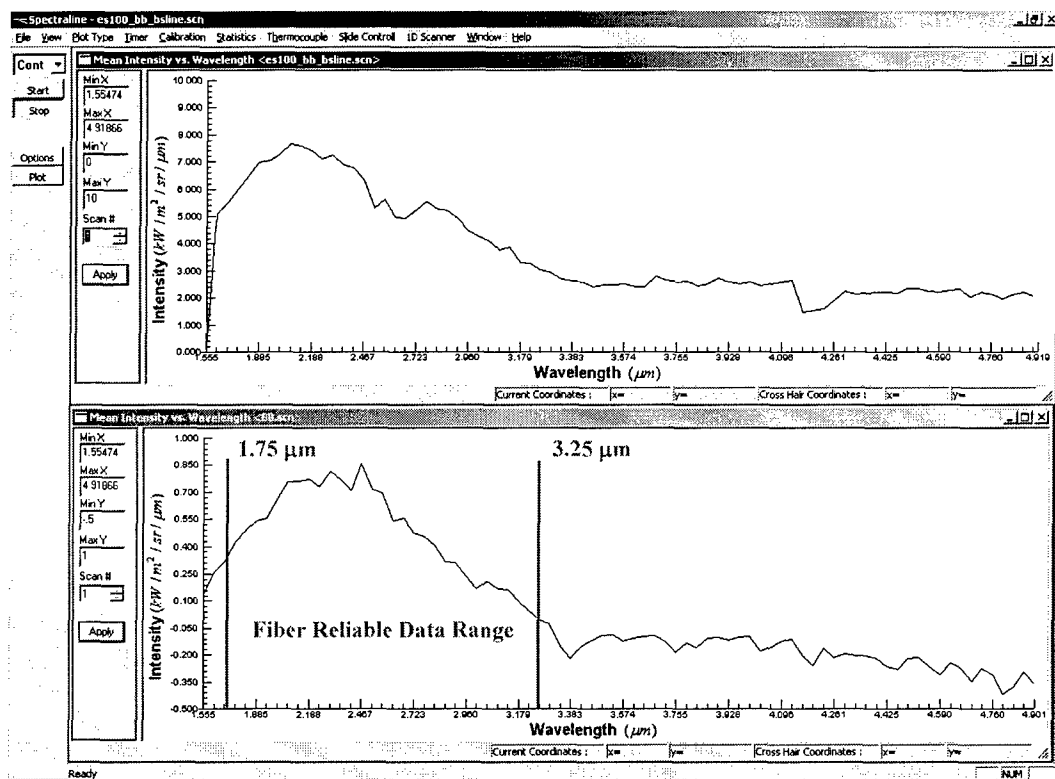
Figure 32.   Blackbody Radiance Measurements for Weighted Filter.

FILTER vs. Wavelength
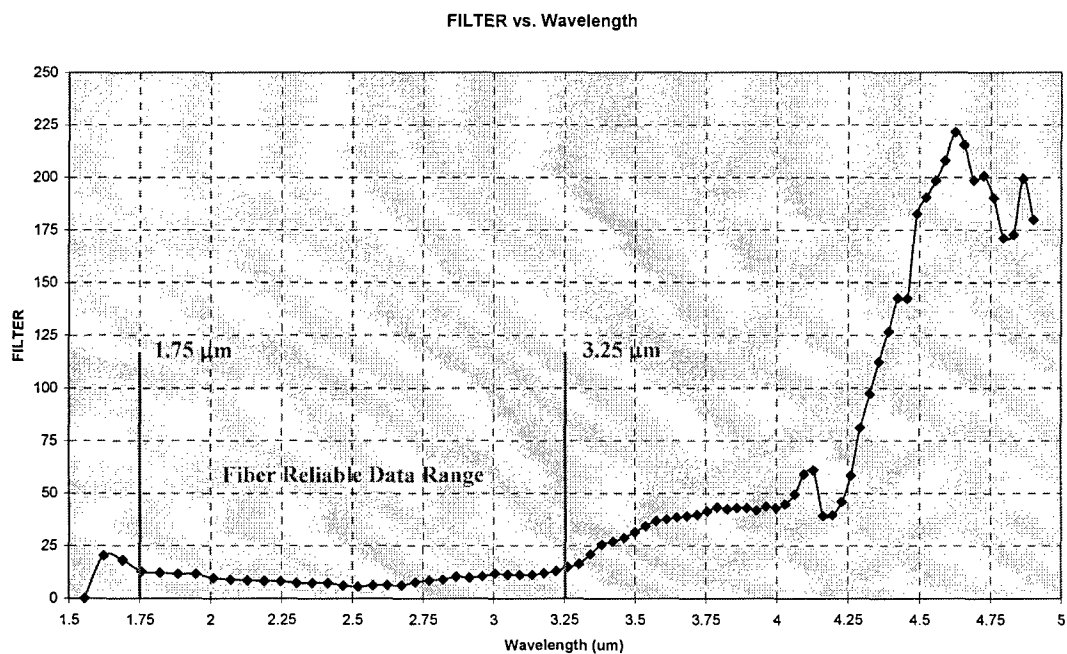
Figure 33.   Weighted Infrared Spectrum Filter.

The filter shows a strong, constant correlation between the blackbody measurements (with and without fiber) from approximately 1.75 to 3.25 μm. This basically agrees with the advertised range of the Thor Labs fiber from 0.5 to 4.0 μm. In this band, the filter "weights" or multiplies the fiber signal by a relatively constant factor range, between 5 and 13. Below 1.75 μm and above 3.25 μm the factor either grows beyond a reasonable value or varies significantly between closely spaced wavelengths. If the signal received by the fiber is too low, the weighting factor becomes disproportionately large, making the data unreliable for the deconvolution code.

Also, before each testing sequence, a background measurement (no exhaust plume or blackbody) was recorded through the fiber, shown in Figure 34. The filter spreadsheet calculations are shown in Appendix E. Weighted Infrared Spectrum Filter Calculations.
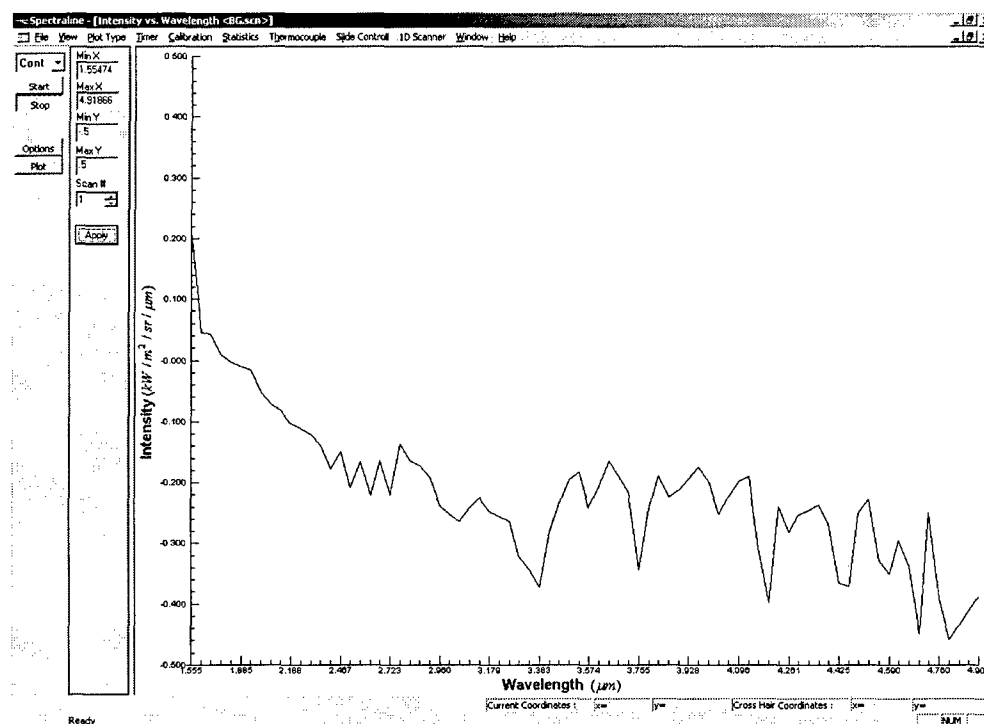
Figure 34. Background Radiance Measurement.

The Input Data Preprocessing Code was run for Runs 1 and 2, creating deconvolution code input files for a particular molecular species at a particular wavelength band. Unfortunately, none of the files converged to a solution in the deconvolution codedue to

43

Rearons are discussed in Section V. Five separate input files were attempted (see Appendix D). After each input file name there is a brief description of the test conditions with a plot of the measured radiance and transmission at each zone. The combustion species CO_2 (primary MIR band at 4.3 μm) and CO (primary MIR band at 4.7 μm) could not be considered since their primary MIR bands lay outside the upper range of the Thor Labs fiber (but within the range of the first, now broken Oriel fiber). The combustion species CH_4 (primary MIR band at 3.3 μm) was also attempted with H_2O with the data obtained from Run 1, but could not converge. The radiance and transmission data is the same, but the wavelength band considered for the particular combustion species and the estimated SPF conditions change.[7]

1) 12X30DA000_H2O.dat – 1.5 O/F mixture ratio, 3.0 exit diameters downstream location, 000 degrees rotation angle (scanned from the top of the plume at 1.2 cm radius to the centerline), looking at the MIR H_2O band (2.43-3.03 μm) centered at 2.7 μm. CO_2 was the secondary species (see Figure 35).

2) 12X30DA000_CH4.dat – 1.5 O/F mixture ratio, 3.0 exit diameters downstream location, 000 degrees rotation angle (scanned from the top of the plume at 1.2 cm radius to the centerline), looking at the MIR CH_4 band (3.0-3.6 μm) centered at 3.3 μm. H_2O was the secondary species (see Figure 35).
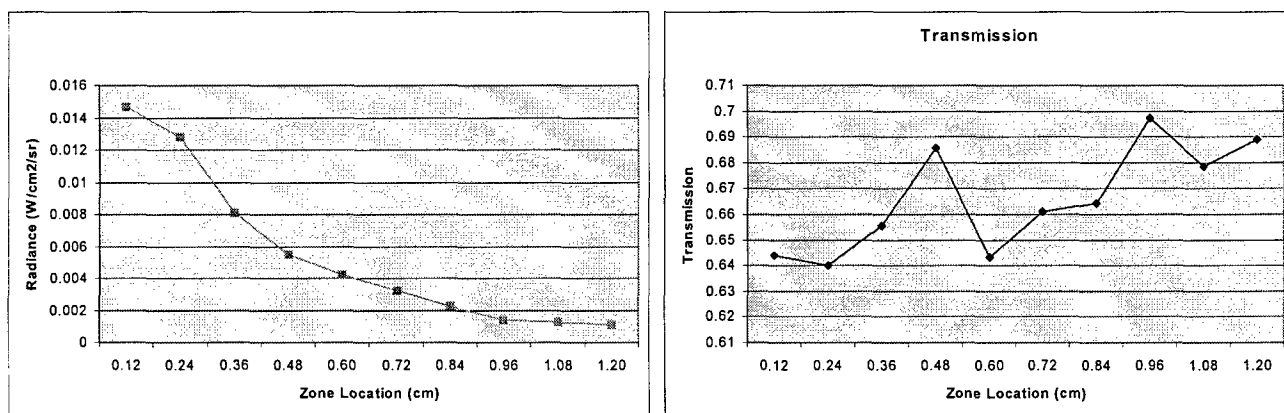
Figure 35.   12X30DA000_XXX Radiance and Transmission.

3) 12X30DA180_H2O.dat – 1.5 O/F mixture ratio, 3.0 exit diameters downstream location, 180 degrees rotation angle (scanned from the centerline of the plume to the bottom at 1.2 cm radius, the rotary stage did not actually move), looking at the MIR H_2O band (2.43-3.03 μm) centered at 2.7 μm. CO_2 was the secondary species (see Figure 36).

4) 12X30DA180_CH4.dat – 1.5 O/F mixture ratio, 3.0 exit diameters downstream location, 180 degrees rotation angle (scanned from the centerline of the plume to the bottom at 1.2 cm radius, the rotary stage did not actually move), looking at the MIR CH_4 band (3.0-3.6 μm) centered at 3.3 μm. H_2O was the secondary species (see Figure 36).
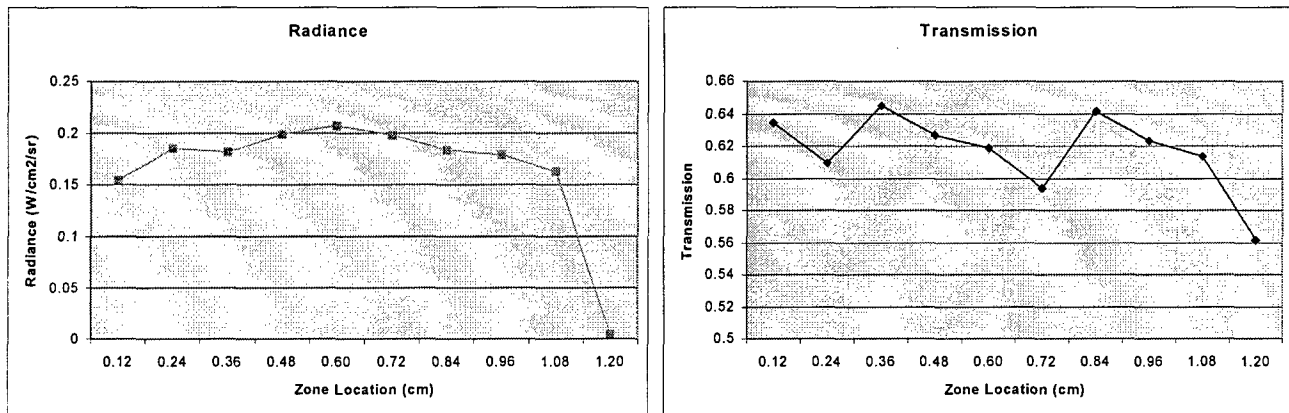
Figure 36.   12X30DA180_XXX Radiance and Transmission.

5) 18X30DA000_H2O.dat – 1.5 O/F mixture ratio, 3.0 exit diameters downstream location, 000 degrees rotation angle (scanned from the top of the plume at 1.8 cm radius to the centerline), looking at the MIR H_2O band (2.43-3.03 μm) centered at 2.7 μm. CO_2 was the secondary species (see Figure 37).

45
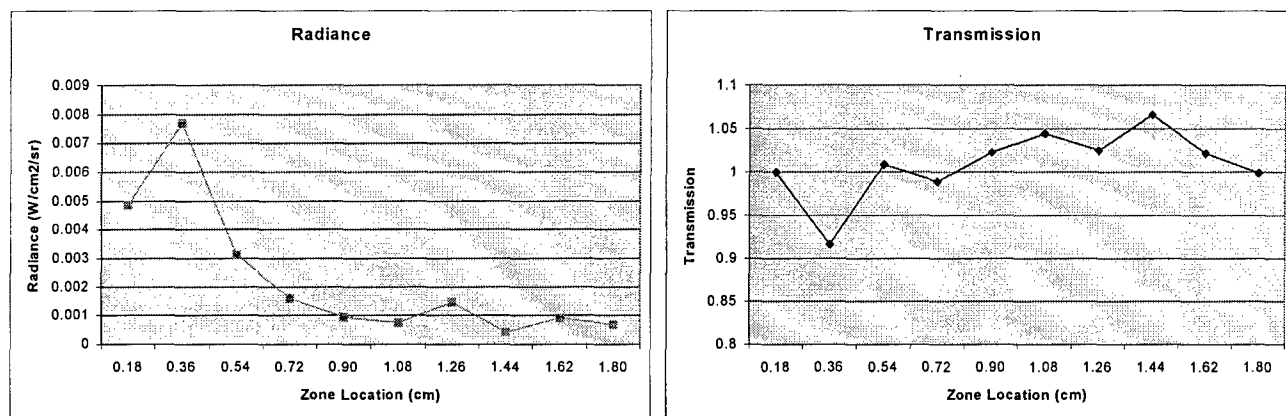
Figure 37.   18X30DA000_XXX Radiance and Transmission.

The combustion species CO_2 (primary MIR band of 4.3 μm) and CO (primary MIR band of 4.7 μm) could not be considered since their primary MIR bands lay outside the upper range of the Thor Labs fiber (but within the range of the broken Oriel fiber).[7]

The radiance and transmission plots should show a trend similar to that discovered in the original work conducted by Dr. Brewer and Dr. Limbaugh for AEDC, referenced in Section II.A.   Looking at a hydrogen-air flame, they measured the following radiance and transmission (see Figure 38).   Logically, at the centerline of the plume, radiance was high and transmission was low, while at the edge of the plume, radiance was low while transmission was high.
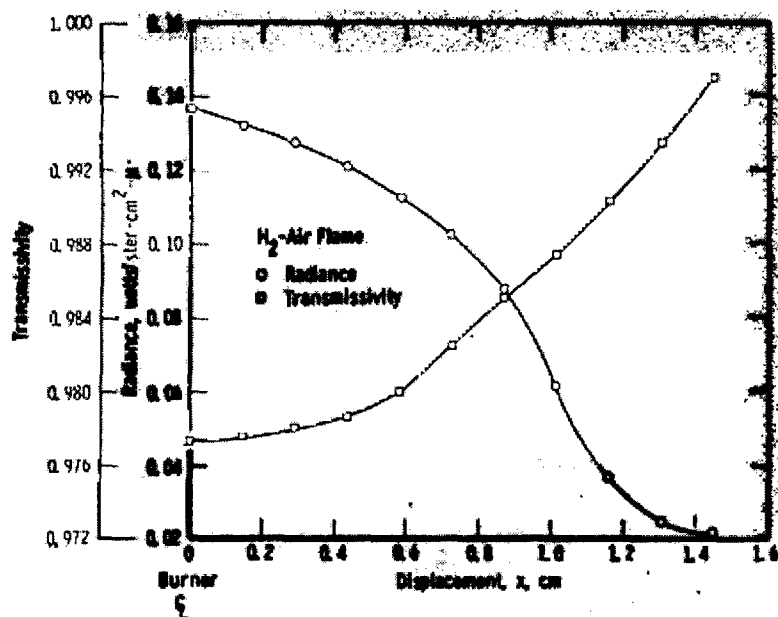
Figure 38.   Original AEDC Radiance and Transmission for H_2-Air Flame.

Using this data for their input file resulted in convergence of the deconvolution code that estimated the following H_2O pressures or concentrations and temperatures at 2.7 μm (see Figure 39).
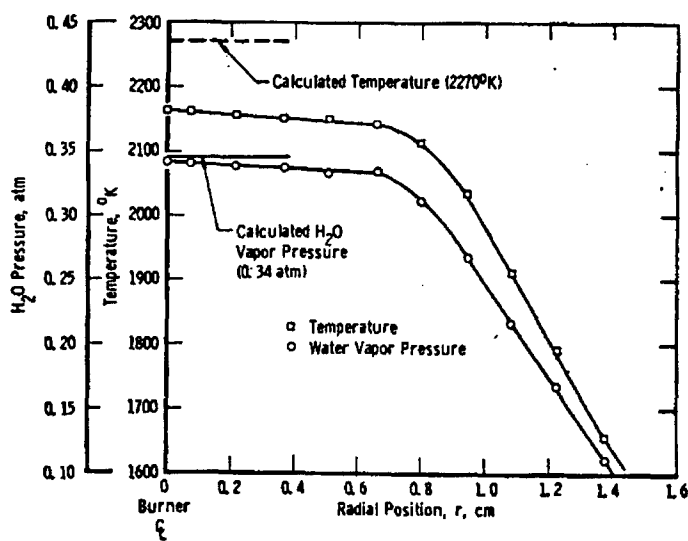
Figure 39.   Original AEDC H_2O Concentrations and Temperatures at 2.7 μm Band.

47

In general, the five radiance curves follow the correct profile, while the transmission curves show an unreliable trend. This data alone could explain why the deconvolution code cannot converge to a solution. The scans from the top to the centerline of the plume (12X30D000_XXX and 18X30D000_XXX) show more correct trends than the scans from the centerline to the bottom of the plume (12X30D180_XXX). This could be the result of mixing irregularities from the rocket motor, or the diagnostic equipment somehow adversely reacting to the testing environment halfway through the firing.

V. CONCLUSIONS

This thesis demonstrated the first time use of fiber optic cable to transmit infrared emission / absorption (E/A) spectra from a rocket plume to an infrared detector. This new diagnostic apparatus technique provides a significant benefit to those interested in measuring rocket exhaust plume characteristics. Expensive, sensitive imaging equipment can be located safely away from the hazardous testing environment (heat, vibration, soot, anomalous explosions, etc.). The new fiber optic configuration allows for rapid translation and rotation around the rocket plume, establishing the capability for rapid spatial characterization of the combustion species present. This thesis demonstrated an entire translation / rotation fiber optic cable scanning mechanism could be constructed from COTS (commercial off the shelf) hardware with minor modifications. The mechanism can be scaled up to scan larger plumes, as long as the hardware can survive the testing environment.

Although the deconvolution code would not converge with the intensity measured through the fiber optic cable, the technique is still viable. Before it broke in preliminary measurements, the original 600 µm core Oriel fiber coupled greater than three times the light intensity compared to the 160 µm core Thor Labs fiber used for this research. An inordinate amount of effort was expended to increase the intensity through the smaller core fiber. Methods attempted were:

1) Adjusting the Fiber Optic Cable Collimator position, angle and FL

2) Adjusting the Fiber Optic Cable Mount position, angle and FL

3) Adjusting the blackbody position and angle

4) Adjusting the Blackbody Intensity Collimator position, angle and FL

5) Adjusting the shutter position and angle

6) Adjusting the Spectraline diagnostic software

7) Rotating the fiber core tip

The signal increased most dramatically when the fiber core tip was rotated, indicating the cleaving process of the tip was not perfectly symmetrical. Ultimately, the signal to noise ratio obtained through the 160 μm core fiber was too low to overcome. The weighted filter was multiplying "noise" by factors of 10 and greater in the primary measurement zone. With the larger intensity received from the 600 μm core fiber, the signal to noise ratio would be much improved and the weighted filter would average a value of 5, estimating from earlier measurements. This improvement alone would significantly increase the quality of the radiance and transmission measurements, allowing the deconvolution code to most likely converge. Experimental results may then be compared to DoD rocket plume model predictions to highlight areas for improvement.

VI. FUTURE WORK

A. SHORT TERM

The most immediate solution to coupling more intensity into the fiber optic cable to reach the spectrometer detector is to procure a larger core fiber. Since the Oriel 600 μm core fiber was the last item of a discontinued lot, a custom built fiber must be ordered. Requesting a 1000 μm core fiber would significantly increase the intensity coupled, and provide the greatest opportunity for the deconvolution code to converge.

While the custom fiber is manufactured, there are several other pieces of the scanning system that can be addressed. A 2 mm aperture can be used for the Fiber Optic Cable Mount. A 1 mm aperture produced a narrow field of view so the fiber only saw the plume and blackbody. Conveniently, a 1 mm aperture dictated the largest size sapphire ball (4.76 mm for the 0.2 NA Oriel fiber) that would fit in a modified SMA connector. A ball twice the size could be used for a 2 mm hole, but a custom fiber connector must be produced. The PC used in the lab can only write files to the harddrive at 2 Hz, or twice a second. This limits how fast we could scan the plume, requiring longer run times. For any test that would involve rotating and translation scanning, a faster computer is required or the Spectraline code could be reprogrammed. The rotary stage can only safely rotate at 1 degree per second with the 1-amp card in the ESP300 controller. Ordering a 2-amp card would allow the stage to rotate at its maximum speed of 80 degrees per second, allowing rotation during testing. A new L-shaped bracket holding the scanning mechanism from the side, as well as some modifications to the rocket motor stand and water cooling lines could alleviate most rotation and translation clearance issues, from virtually any downstream location.

B. LONG TERM

The AEDC combustion species deconvolution code does not account for the advances in imaging spectrometers in the past 30 years. A new code, that could analyze all the radiation and transmission data provided by the scanning mechanism (different slices at multiple angles) over all wavelengths visible to the spectometer to the fiber, not just averaged, integrated values, would produce a true nonsymmetric combustion species map of the exhaust plume. Such a code is most likely PhD level work.

51

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. STANDARD PLUME FLOWFIELD (SPF) MODEL

Frame 001 | 14 Dec 2001 | sd = 1.2 well mixed engine results

NPS Engine
RP1/GOX
P_c = 2.51 MPa
A_t = 762 mm
A_e/A_t = 4.1

CO2
0.05
0.0476554
0.0453108
0.0429662
0.0406216
0.0382769
0.0359323
0.0335877
0.0312431
0.0288985
0.0265539
0.0242093
0.0218647
0.0195201
0.0171754
0.0148308
0.0124862
0.0101416
0.00779699
0.00545238

R (cm)

X (cm)

Frame 001 | 14 Dec 2001 | sd = 1.2 well mixed engine results

NPS Engine
RP1/GOX
P_c = 2.51 MPa
A_t = 762 mm
A_e/A_t = 4.1

H2O (mole fraction)
0.160476
0.152452
0.144429
0.136405
0.128381
0.120357
0.112333
0.10431
0.0962857
0.0882619
0.0802381
0.0722143
0.0641905
0.0561667
0.0481429
0.040119
0.0320952
0.0240714
0.0160476
0.00802381

R (cm)

X (cm)

54

APPENDIX B: INPUT DATA PREPROCESSING C++ CODE

```cpp
// ETdata2.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"

#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <math.h>

#define SPFR 17
#define EXPR 20
#define MAXPIXELS 80

int getpixel(double wl, int minmax);
void exiterror(char error[256]);
FILE *openfile(char fname[80], char rw[5]);
void closefile(FILE *fid, char fname[80]);
double interpol(double x_value, const double x[SPFR], const double y[SPFR]);

void main(int argc, char* argv[])
{
            int   i, j, k, pixel_min, pixel_max, x_location, dAngle, spec1, spec2, maxfilt;
            double BG[MAXPIXELS], BB[MAXPIXELS], ref_BG[MAXPIXELS], ref_BB[MAXPIXELS];
            double dWL, dE, dPress, dTemp, dR, r, inc;
            double WL_min, WL_max;
            double E[EXPR][MAXPIXELS], T[EXPR][MAXPIXELS], WL[MAXPIXELS], Trans[EXPR], Emiss[EXPR];
            double SPF_R[SPFR], SPF_P[SPFR], SPF_T[SPFR];
            double H2O[SPFR], CO2[SPFR], CH4[SPFR], CO[SPFR], OH[SPFR], NO[SPFR], Pri[SPFR], Sec[SPFR];
            double dH2O, dCO2, dCH4, dCO, dOH, dNO;
            double FILTER[MAXPIXELS];
            double Pout[EXPR], Tout[EXPR], Secout[EXPR], Priout[EXPR];
            char  basefilename[80], filename[80], buff[256], output1[80], output2[80];
            char  background[80], blackbody[80], ref_background[80], ref_blackbody[80];
            FILE *fid, *fout1, *fout2;

//          Initializing variables
            for (i=0;i<EXPR;i++) Emiss[i]=Trans[i]=0.0;

//          User input
            if (1 == argc)
            {
                      printf("Enter the REFERENCE BACKGROUND intensity filename: ");
                      scanf("%s",ref_background);
                      printf("Enter the REFERENCE BLACKBODY intensity filename: ");
                      scanf("%s",ref_blackbody);
                      printf("Enter the BACKGROUND intensity filename: ");
                      scanf("%s",background);
                      printf("Enter the BLACKBODY intensity filename: ");
                      scanf("%s",blackbody);
                      printf("Enter the base_filename for the ES100 data (i.e. \"X05DA045_\"): ");
                      scanf("%s",basefilename);
                      printf("Enter the min Wavelength (1.58 < um < 4.90): ");
                      scanf("%lf",&WL_min);
                      printf("Enter the max Wavelength (%3lf < um < 4.90): ",(WL_min+0.01));
                      scanf("%lf",&WL_max);
                      printf("\nDownstream Location\n #\tLocation\n");
                      printf("---------------\n1:\t0.5D\n2:\t1.0D\n3:\t1.5D\n4:\t2.0D\n5:\t3.0D\n6:\t4.0D\n");
                      printf("---------------\nPlease select downstream location# (1-6): ");
                      scanf("%d", &x_location);
                      printf("\n\nPrimary Species\n---------------\n1: H2O\n2: CH4\n3: CO2\n4: CO\n");
                      printf("---------------\nPlease select species # (1-4): ");
                      scanf("%d", &spec1);
```

```c
                            printf("\n\nSecondary Species\n---------------\n0: NONE\n1: H20\n2: CH4\n3: CO2\n4: CO\n5:
NO\n6: OH\n");
                            printf("---------------\nPlease select species # (0-6): ");
                            scanf("%d", &spec2);
                            fflush(stdin);
                    }
                    else if (9 == argc)
                    {
                            strcpy(background,argv[1]);
                            strcpy(blackbody,argv[2]);
                            strcpy(basefilename,argv[3]);
                            WL_min=(double)atof(argv[4]);
                            WL_max=(double)atof(argv[5]);
                            x_location=atoi(argv[6]);
                            spec1=atoi(argv[7]);
                            spec2=atoi(argv[8]);
                            printf("argv[]      =       %s      %s      %s      %lf     %lf     %d      %d
%d\n",background,blackbody,basefilename,WL_min,WL_max,x_location,spec1,spec2);
                    }
                    else
                    {
                            exiterror("USE:\tETdata.exe needs 8 arguments for non-interactive mode\n\t\tor ZERO arguments
otherwise\n");
                    }


                    if (spec2 == spec1)     exiterror("Primary and Secondary species cannot be the same\n");

//          Inform user of code assumptions
            if (1 == argc)
            {
                    printf("\n\nThis code assumes:\n1: the even number filenames are Transmission Data.\n");
                    printf("2: the odd number filenames are Emission Data.\n\n");
            }
            k=0;

//          Here we read the background intensity
            fid=openfile(ref_background,"rt");
            for (i=0;i<MAXPIXELS;i++)
            {
                    //Ignore the first line of each file (labels only!)
                    if (0==i)  fscanf(fid,"%*s %*s %*s %*s %*s %*s %*s\n");

                    fscanf(fid,"%*d %lf %*lf %*lf %*lf %lf %*lf\n", &dWL, &dE);
                    ref_BG[i]=dE;
                    WL[i]=dWL;
            }
            closefile(fid,background);

//          Here we read the blackbody intensity
            fid=openfile(ref_blackbody,"rt");
            for (i=0;i<MAXPIXELS;i++)
            {
                    //Ignore the first line of each file (labels only!)
                    if (0==i)  fscanf(fid,"%*s %*s %*s %*s %*s %*s %*s\n");

                    fscanf(fid,"%*d %*lf %*lf %*lf %*lf %lf %*lf\n", &dE);
                    ref_BB[i]=dE;
            }
            closefile(fid,blackbody);

//          Here we read the background intensity
            fid=openfile(background,"rt");
            for (i=0;i<MAXPIXELS;i++)
            {
                    //Ignore the first line of each file (labels only!)
                    if (0==i)  fscanf(fid,"%*s %*s %*s %*s %*s %*s %*s\n");

                    fscanf(fid,"%*d %*lf %*lf %*lf %*lf %lf %*lf\n", &dE);
                    BG[i]=dE;
```

```
                }
                closefile(fid,background);

//              Here we read the blackbody intensity
                fid=openfile(blackbody,"rt");
                for (i=0;i<MAXPIXELS;i++)
                {
                        //Ignore the first line of each file (labels only!)
                        if (0==i)  fscanf(fid,"%*s %*s %*s %*s %*s %*s %*s\n");

                        fscanf(fid,"%*d %*lf %*lf %*lf %*lf %lf %*lf\n", &dE);
                        BB[i]=dE;
                }
                closefile(fid,blackbody);

//              This loop reads all 40 files, 20 Emission data and 20 Transmission data files.
                for(i=0;i<(2*EXPR);i++)
                {
                        itoa(i,buff,10);
                        strcpy(filename,basefilename);
                        strcat(filename,buff);
                        strcat(filename,".txt");

                        fid=openfile(filename,"rt");

//                      This loop reads the contents of one file into memory (there are 160/2 pixels in the ES100)
                        for(j=0;j<MAXPIXELS;j++)
                        {
                                if (0==j) //Ignore the first line of each file (labels only!)
                                {
                                        fscanf(fid,"%*s %*s %*s %*s %*s %*s %*s\n");
                                }

//                              fscanf(fid,"%*d %*lf %*lf %*lf %*lf %lf %lf\n", &dE, &dT);
                                fscanf(fid,"%*d %*lf %*lf %*lf %*lf %lf %*lf\n", &dE);


                                if (0 == (fmod(i,2)) )   //Transmission data comes from Even # files
                                {
                                        T[k][j]=dE;                                     //workout   trans   eq   w/   -BG
everywhere.
                                }
                                else                                            //Emission data comes from Odd
# files
                                {
                                        E[k][j]=dE;
                                }
                        }

                        if ( (fmod(i,2)) > 0)  k++;           //Only increment k after reading an odd file (kmax=20)

                        closefile(fid,filename);
                }

// These loops calculate the transmission = [(P + BB) - P]/BB = [evenfiles - odd]/BB
                for(i=0;i<EXPR;i++)
                {
                        for(j=0;j<MAXPIXELS;j++)
                        {
                                T[i][j] = (T[i][j] - E[i][j])/(BB[j]);
                        }
                }

//Now we can subtract the Background.

                for(i=0;i<EXPR;i++)
                {
                        for(j=0;j<MAXPIXELS;j++)
                        {
                                E[i][j] -= BG[j];
```

```
                    }
            }


//          Calculating & writing the filter for the ES100 w/ fiber-optic waveguide.

            fid=openfile("FILTER.dat","wt");

            for(i=0;i<MAXPIXELS;i++)
            {
                    FILTER[i] = (ref_BB[i] - ref_BG[i])/(BB[i] - BG[i]);
                    fprintf(fid,"%lf %lf\n",WL[i],FILTER[i]);
            }
            closefile(fid,"FILTER.dat");

//          Calculating the index corresponding to the desired wavelengths.

            pixel_min=getpixel(WL_min,0);
            pixel_max=getpixel(WL_max,1);
            maxfilt=pixel_max - pixel_min + 1;

//          This loop integrates the data over the specified wavelengths
            for (i=0;i<EXPR;i++)
            {
                    for (j=pixel_min;j<=pixel_max;j++)
                    {
                            dWL = WL[j+1]-WL[j];
                            Trans[i]+=T[i][j]*dWL;
                            Emiss[i]+=E[i][j]*dWL;
                    }
            }


//          Reading data from SPF file...

            switch (x_location)
            {
                    case 1:
                            strcpy(filename,"X05D_SPF.dat");
                            strcpy(output1,basefilename);
                            strcat(output1,"AEDC.dat");
                            strcpy(output2,"X05DA");
                            break;
                    case 2:
                            strcpy(filename,"X10D_SPF.dat");
                            strcpy(output1,basefilename);
                            strcat(output1,"AEDC.dat");
                            strcpy(output2,"X10DA");
                            break;
                    case 3:
                            strcpy(filename,"X15D_SPF.dat");
                            strcpy(output1,basefilename);
                            strcat(output1,"AEDC.dat");
                            strcpy(output2,"X15DA");
                            break;
                    case 4:
                            strcpy(filename,"X20D_SPF.dat");
                            strcpy(output1,basefilename);
                            strcat(output1,"AEDC.dat");
                            strcpy(output2,"X20DA");
                            break;
                    case 5:
                            strcpy(filename,"X30D_SPF.dat");
                            strcpy(output1,basefilename);
                            strcat(output1,"AEDC.dat");
                            strcpy(output2,"X30DA");
                            break;
                    case 6:
                            strcpy(filename,"X40D_SPF.dat");
                            strcpy(output1,basefilename);
```

```c
                        strcat(output1,"AEDC.dat");
                        strcpy(output2,"X40DA");
                        break;
                default:
                        printf("I'm in default & filename = %s\n",filename);
                        break;
        }

//        makes the output2 filename
        for (i=5;i<9;i++)
                        buff[i-5]=basefilename[i];

        buff[4]='\0';
        dAngle=atoi(buff);
        dAngle += 180;
        itoa(dAngle,buff,10);
        strcat(output2,buff);
        strcat(output2,"_AEDC.dat");

        fid=openfile(filename,"rt");

//        There are 17 text lines that we can skip
        for (i=0;i<17;i++) fgets(buff,256,fid);

//        Reading all SPF data
        for (i=0; i<SPFR; i++)
        {
                fscanf(fid,"%*lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %*lf %*lf %*lf %*lf %*lf %*lf \n", \
                        &dR, &dTemp, &dPress, &dH2O, &dCO2, &dCH4, &dCO, &dOH, &dNO);

                SPF_R[i]= dR;                   // cm
                SPF_T[i]= dTemp;      // K
                SPF_P[i]= dPress;     // atm
                H2O[i]  = dH2O;                 // mole fraction
                CO2[i]  = dCO2;                 // mole fraction
                CH4[i]  = dCH4;                 // mole fraction
                CO[i]   = dCO;                  // mole fraction
                OH[i]   = dOH;                  // mole fraction
                NO[i]   = dNO;                  // mole fraction
        }

        closefile(fid,filename);

// Selecting Primary and Secondary species...
        switch (spec1)
        {
                case 1:
                        for (i=0;i<SPFR;i++)
                                Pri[i]=H2O[i];
                        break;
                case 2:
                        for (i=0;i<SPFR;i++)
                                Pri[i]=CH4[i];
                        break;
                case 3:
                        for (i=0;i<SPFR;i++)
                                Pri[i]=CO2[i];
                        break;
                case 4:
                        for (i=0;i<SPFR;i++)
                                Pri[i]=CO[i];
                        break;
                default:
                        exiterror("Primary species out of range ");
                        break;
        }

        switch (spec2)
        {
                case 0:
```

59

```c
                for (i=0;i<SPFR;i++)
                        Sec[i]=0.0;
                break;
        case 1:
                for (i=0;i<SPFR;i++)
                        Sec[i]=H2O[i];
                break;
        case 2:
                for (i=0;i<SPFR;i++)
                        Sec[i]=CH4[i];
                break;
        case 3:
                for (i=0;i<SPFR;i++)
                        Sec[i]=CO2[i];
                break;
        case 4:
                for (i=0;i<SPFR;i++)
                        Sec[i]=CO[i];
                break;
        case 5:
                for (i=0;i<SPFR;i++)
                        Sec[i]=NO[i];
                break;
        case 6:
                for (i=0;i<SPFR;i++)
                        Sec[i]=OH[i];
                break;
        default:
                exiterror("Secondary species out of range");
                break;
}

switch (x_location)
{
        case 1:
                r=0.900;
                inc=0.090;
                break;
        case 2:
                r=0.950;
                inc=0.0950;
                break;
        case 3:
                r=0.950;
                inc=0.0950;
                break;
        case 4:
                r=0.950;
                inc=0.0950;
                break;
        case 5:
                r=1.800;
                inc=0.180;
                break;
        case 6:
                r=1.500;
                inc=0.150;
                break;
}

r -=inc;

// Interpolating...
        for( i=1, j=EXPR-2; i<j; i++, j--)
        {
                Pout[i]=interpol(r,SPF_R,SPF_P);
                Tout[i]=interpol(r,SPF_R,SPF_T);
                Secout[i]=interpol(r,SPF_R,Sec);
                Priout[i]=interpol(r,SPF_R,Pri);
```

60

```
                                r -= inc;

                                Tout[j]=Tout[i];
                                Pout[j]=Pout[i];
                                Secout[j]=Secout[i];
                                Priout[j]=Priout[i];
                }

//              For the edges, (i.e. R = inc and R = Max_R, we do not interpolate,
//              instead just take the SPF min & max values

                Pout[0] = SPF_P[SPFR - 1];
                Tout[0] = SPF_T[SPFR - 1];
                Secout[0] = Sec[SPFR - 1];
                Priout[0] = Pri[SPFR - 1];
                Pout[EXPR/2-1] = SPF_P[0];
                Tout[EXPR/2-1] = SPF_T[0];
                Secout[EXPR/2-1] = Sec[0];
                Priout[EXPR/2-1] = Pri[0];
                Pout[EXPR/2] = SPF_P[0];
                Tout[EXPR/2] = SPF_T[0];
                Secout[EXPR/2] = Sec[0];
                Priout[EXPR/2] = Pri[0];
                Pout[EXPR-1]=Pout[0];
                Tout[EXPR-1]=Tout[0];
                Secout[EXPR-1]=Secout[0];
                Priout[EXPR-1]=Priout[0];


//              Writing output...
                fout1=openfile(output1,"wt");
                fout2=openfile(output2,"wt");

                fprintf(fout1,"2 2\n10 0. -1 1 0\n%s\n",output1);
                fprintf(fout2,"2 2\n10 0. -1 1 0\n%s\n",output2);

                r=inc*10;
                for (i=0, j=EXPR-1; i<j; i++, j--)
                {
                                fprintf(fout1,"%lf            %lf          %lf          %lf          %lf          %lf
%lf\n",r,Pout[i],Tout[i],Secout[i],Priout[i],0.10*Emiss[i],Trans[i]);
                                fprintf(fout2,"%lf            %lf          %lf          %lf          %lf          %lf
%lf\n",r,Pout[j],Tout[j],Secout[j],Priout[j],0.10*Emiss[j],Trans[j]);
                                r -= inc;
                }

                fprintf(fout1,"%d\n",maxfilt);
                fprintf(fout2,"%d\n",maxfilt);

                for (i=pixel_min; i<=pixel_max; i++)
                {
                                fprintf(fout1,"%lf %lf\n", WL[i], FILTER[i]);
                                fprintf(fout2,"%lf %lf\n", WL[i], FILTER[i]);
                }
                closefile(fout1,output1);
                closefile(fout2,output2);

                fprintf(stderr,"\n\n\t\t<press ENTER to END this program>");

                while (10 != fgetc(stdin))           { }

}


int getpixel(double wl, int minmax)
//////////////////////////////////////////////////////////////////////////
//              Given:  a wavelength and a minmax (0 or 1),
//              Returns: the corresponding index number (int) rounded down or up
//                                  with respect to minmax's value (o = round down, 1 = round up.)
//////////////////////////////////////////////////////////////////////////
{
```

```c
                double pixel;

                if ((minmax < 0) && (minmax > 1)) exiterror("getpixel() called with the wrong argument");

        //              pixel=0.041251*pow(wl,6) - 1.0283*pow(wl,5) + 9.2168*pow(wl,4) - 39.355*pow(wl,3) + 91.810*wl*wl -
82.785*wl + 9.6513;
                pixel = -1.12230E-01*pow(wl,5) + 1.42430*pow(wl,4) - 6.57619*wl*wl*wl + 1.64349E+01*wl*wl -
7.09461*wl - 1.12873E+01;
                if (0==minmax)
                                return((int)floor(pixel));
                else
                                return((int)ceil(pixel));
        }

        void exiterror(char error[256])
        ///////////////////////////////////////////////
        //              Given: An eror message (error)
        //              Exits program and sends the error message to stderr.
        ///////////////////////////////////////////////
        {
                fprintf(stderr,"ERROR: %s\n",error);
                fprintf(stderr,"\t\t<press ENTER to exit and close this window>");

                while (10 != fgetc(stdin))              { }

                exit(EXIT_FAILURE);
        }

        double interpol(double x_value, const double x[SPFR], const double y[SPFR])
        /////////////////////////////////////////////////////////////////////////
        //              Given:  an X value, and two arrays X[] and Y[] of the same size
        //              Returns: the interpolated Y value from array Y[] corresponding to the X value
        //              Uses:   Linear interpolation
        /////////////////////////////////////////////////////////////////////////
        {
                int i=0;
                double y_value;
                while(x_value > x[i])
                {
                        i++;
                }
                y_value=y[i-1] + (x_value - x[i-1]) * (y[i] - y[i-1]) / (x[i] - x[i-1]);
                return(y_value);
        }

        FILE *openfile(char fname[80], char rw[5])
        {
                FILE *fid;
                char error_msg[80];

                if ( (fid=fopen(fname,rw)) == NULL)
                {
                        strcpy(error_msg,"Cannot open file: ");
                        strcat(error_msg,fname);
                        strcat(error_msg," for: ");
                        strcat(error_msg,rw);
                        strcat(error_msg,"\n");
                        exiterror( error_msg);
                }
                return (fid);


        }

        void closefile(FILE *fid, char fname[80])
        {
                char error_msg[80];

                if (0 != (fclose(fid)))
                {
```

62

```
                strcpy(error_msg,"While closing filename: ");
                strcat(error_msg,fname);
                strcat(error_msg,"\n");
                exiterror(error_msg);
        }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. INPUT DATA FILES

1) 12X30DA000_H2O.dat

2 2
10 0. -1 1 0
X30DA000_AEDC.dat
1.200000 1.000000 1000.229204 0.000862 0.002149 0.0010823 0.689135
1.080000 1.000000 1000.004623 0.002178 0.007178 0.0012713 0.678541
0.960000 1.000000 1000.191623 0.006820 0.024949 0.0014055 0.697253
0.840000 1.000000 1000.487457 0.015964 0.060065 0.0022650 0.664367
0.720000 1.068298 1187.373204 0.029359 0.110904 0.0032346 0.660906
0.600000 1.126501 1090.706082 0.036047 0.135477 0.0042141 0.643225
0.480000 0.968959 1036.133391 0.036310 0.136400 0.0055026 0.685814
0.360000 0.901642 1018.382639 0.036310 0.136300 0.0081269 0.655649
0.240000 1.020960 1077.083274 0.036308 0.136300 0.0128362 0.640004
0.120000 1.187764 1580.022347 0.036300 0.136300 0.0146586 0.644142
14
2.413130 7.426745
2.467000 6.185777
2.519970 5.616729
2.572070 6.281784
2.623320 6.314058
2.673740 6.304229
2.723340 7.706413
2.772150 8.630134
2.820180 9.012326
2.867450 10.410054
2.913990 9.952110
2.959810 10.458379
3.004940 11.838482
3.049380 11.359359

2) 12X30DA000_CH4.dat

2 2

10 0. -1 1 0

X30DA000_AEDC.dat

1.200000 1.000000 323.229204 0.002149 0.000224 0.0005255 0.682143

1.080000 1.000000 403.004623 0.007178 0.000746 0.0009102 0.610737

0.960000 1.000000 644.191623 0.024949 0.002577 0.0007046 0.693339

0.840000 1.000000 992.487457 0.060065 0.006090 0.0016335 0.576918

0.720000 1.068298 1187.373204 0.110904 0.011338 0.0012042 0.649238

0.600000 1.126501 1090.706082 0.135477 0.014411 0.0015104 0.671028

0.480000 0.968959 1036.133391 0.136400 0.014580 0.0022026 0.673610

0.360000 0.901642 1018.382639 0.136300 0.014580 0.0022322 0.664875

0.240000 1.020960 1077.083274 0.136300 0.014579 0.0043646 0.587820

0.120000 1.187764 1580.022347 0.136300 0.014570 0.0052004 0.624909

17

2.959810 10.458379

3.004940 11.838482

3.049380 11.359359

3.093160 11.185715

3.136300 11.158873

3.178820 12.009839

3.220740 13.049482

3.262080 14.949090

3.302840 16.394192

3.343070 21.057485

3.382770 25.571744

3.421960 26.985917

3.460670 28.771242

3.498900 31.415868

3.536690 34.288582

3.574050 36.811413

3.611000 37.527172

3) 12X30DA180_H2O.dat

2 2

10 0. -1 1 0

X30DA180_AEDC.dat

1.200000 1.000000 1000.229204 0.000862 0.002149 0.004308 0.561395

1.080000 1.000000 1000.004623 0.002178 0.007178 0.163220 0.613875

0.960000 1.000000 1000.191623 0.006820 0.024949 0.179317 0.623679

0.840000 1.000000 1000.487457 0.015964 0.060065 0.184300 0.641228

0.720000 1.068298 1187.373204 0.029359 0.110904 0.198534 0.593728

0.600000 1.126501 1090.706082 0.036047 0.135477 0.207643 0.618632

0.480000 0.968959 1036.133391 0.036310 0.136400 0.199619 0.627205

0.360000 0.901642 1018.382639 0.036310 0.136300 0.181926 0.645026

0.240000 1.020960 1077.083274 0.036308 0.136300 0.185632 0.609315

0.120000 1.187764 1580.022347 0.036300 0.136300 0.154473 0.634240

14

2.413130 7.426745

2.467000 6.185777

2.519970 5.616729

2.572070 6.281784

2.623320 6.314058

2.673740 6.304229

2.723340 7.706413

2.772150 8.630134

2.820180 9.012326

2.867450 10.410054

2.913990 9.952110

2.959810 10.458379

3.004940 11.838482

3.049380 11.359359

4) 12X30DA180_CH4.dat

2 2

10 0. -1 1 0

X30DA180_AEDC.dat

1.200000 1.000000 323.229204 0.002149 0.000224 0.0006713 0.626215

1.080000 1.000000 403.004623 0.007178 0.000746 0.0045913 0.699235

0.960000 1.000000 644.191623 0.024949 0.002577 0.0057445 0.594474

0.840000 1.000000 992.487457 0.060065 0.006090 0.0062872 0.567703

0.720000 1.068298 1187.373204 0.110904 0.011338 0.0064067 0.612163

0.600000 1.126501 1090.706082 0.135477 0.014411 0.0063176 0.647754

0.480000 0.968959 1036.133391 0.136400 0.014580 0.0069079 0.546998

0.360000 0.901642 1018.382639 0.136300 0.014580 0.0052952 0.716642

0.240000 1.020960 1077.083274 0.136300 0.014579 0.0064529 0.544082

0.120000 1.187764 1580.022347 0.136300 0.014570 0.0060542 0.588896

17

2.959810 10.458379

3.004940 11.838482

3.049380 11.359359

3.093160 11.185715

3.136300 11.158873

3.178820 12.009839

3.220740 13.049482

3.262080 14.949090

3.302840 16.394192

3.343070 21.057485

3.382770 25.571744

3.421960 26.985917

3.460670 28.771242

3.498900 31.415868

3.536690 34.288582

3.574050 36.811413

3.611000 37.527172

5) 18X30DA000_H2O.dat

2 2

10 0. -1 1 0

X30DA000_AEDC.dat

1.800000 1.000000 323.229204 0.000862 0.002149 0.000696 0.999257

1.620000 1.000000 323.229204 0.000862 0.002149 0.000911 1.021953

1.440000 1.000000 323.229204 0.000862 0.002149 0.000427 1.066337

1.260000 1.000000 323.229204 0.000862 0.002149 0.001461 1.024595

1.080000 1.000000 403.004623 0.002178 0.007178 0.000766 1.044443

0.900000 1.000000 812.312566 0.010580 0.039384 0.000945 1.022529

0.720000 1.068298 1187.373204 0.029359 0.110904 0.001602 0.988954

0.540000 1.031525 1055.727328 0.036288 0.136323 0.003123 1.008805

0.360000 0.901642 1018.382639 0.036310 0.136300 0.007707 0.915983

0.180000 1.187764 1580.022347 0.036300 0.136300 0.004850 1.000042

14

2.413130 8.049628

2.467000 6.598992

2.519970 6.220666

2.572070 6.693529

2.623320 7.001140

2.673740 7.013476

2.723340 8.246711

2.772150 9.364367

2.820180 9.706529

2.867450 10.923651

2.913990 10.693245

2.959810 11.002759

3.004940 11.579382

3.049380 10.885821

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. WEIGHTED INFRARED SPECTRUM FILTER CALCULATIONS

ES100 (blkbdy - bkgrnd)	Fiber (blkbdy - bkgrnd)	Wavelength	FILTER (ES100/Fiber)
-0.013398	-0.073485	1.55474	0.182322923
5.0377762	0.2482661	1.62309	20.29184089
5.4545473	0.3004146	1.69028	18.15673173
5.95025241	0.46286488	1.75634	12.85526871
6.47908903	0.53122277	1.82127	12.19655744
6.9665099	0.5981395	1.8851	11.64696513
7.051548	0.6076526	1.94784	11.60457143
7.3028561	0.7553164	2.00952	9.668605236
7.713222	0.8602047	2.07016	8.966728501
7.6485488	0.8859949	2.12978	8.632723281
7.4894648	0.8980775	2.18838	8.339441529
7.218691	0.883161	2.246	8.173697661
7.356624	0.985305	2.30265	7.466341894
7.022772	0.946983	2.35836	7.415943053
6.906227	0.929913	2.41313	7.426745298
6.495276	1.050034	2.467	6.185776842
5.481079	0.975849	2.51997	5.616728613
5.770371	0.918588	2.57207	6.281783563
5.154519	0.816356	2.62332	6.314057838
5.133275	0.814259	2.67374	6.304228753
5.438978	0.705773	2.72334	7.706412685
5.703293	0.660858	2.77215	8.630133856
5.425348	0.601992	2.82018	9.012325745
5.371869	0.516027	2.86745	10.41005413
5.063305	0.508767	2.91399	9.952109708
4.685657	0.448029	2.95981	10.45837881
4.491295	0.379381	3.00494	11.83848163
4.325394	0.380778	3.04938	11.359359
3.998278	0.357445	3.09316	11.18571528
4.094302	0.36691	3.1363	11.15887275
3.535119	0.2943519	3.17882	12.00983924
3.525697	0.27017908	3.22074	13.04948185
3.284632	0.2197212	3.26208	14.94909003
3.176801	0.193776	3.30284	16.39419226
3.041859	0.144455	3.34307	21.05748503
3.015957	0.117941	3.38277	25.5717435
2.872327	0.106438	3.42196	26.98591668
2.659182	0.092425	3.46067	28.77124155
2.669532	0.084974	3.4989	31.41586838
2.656165	0.077465	3.53669	34.28858194
2.743481	0.074528	3.57405	36.81141316
2.598494	0.069243	3.611	37.52717242
2.586601	0.06742	3.64756	38.36548502
2.94665	0.075183	3.68374	39.19303566
2.867652	0.072223	3.71957	39.70552317
2.835912	0.068361	3.75507	41.48435511
2.828876	0.065309	3.79026	43.31525517
2.67054	0.062899	3.82515	42.45759074
2.729127	0.063534	3.85976	42.95537822
2.904528	0.067498	3.89412	43.03131945
2.776746	0.066316	3.92823	41.87143374
2.695576	0.061772	3.96213	43.63750567
2.745617	0.064039	3.99584	42.8741392
2.694596	0.060083	4.02936	44.84789375
2.744816	0.055456	4.06272	49.49538373
2.767112	0.046781	4.09594	59.15033881
2.818051	0.046174	4.12904	61.03112141
1.722612	0.043821	4.16203	39.31019374
1.848248	0.046591	4.19494	39.66963577
1.819121	0.039655	4.22779	45.87368554
2.184446	0.03731	4.26059	58.54853927
2.488351	0.030646	4.29336	81.19659988
2.397659	0.024661	4.32613	97.2247273
2.393115	0.021291	4.35891	112.40031
2.478961	0.019564	4.39172	126.7103353
2.514091	0.017631	4.42458	142.594918
2.500722	0.017569	4.45751	142.3371848
2.569844	0.01407	4.49054	182.6470505
2.583917	0.013569	4.52367	190.4279608
2.545123	0.012825	4.55692	198.4501365
2.570126	0.012359	4.59033	207.9558217
2.550326	0.011511	4.6239	221.5555556
2.593563	0.012034	4.65765	215.5196111

71

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Scruton, Zachary M., *Measurement of Fuel Additive Effects on the Soot Mass Loading in Oxygen/Kerosene Exhaust Plumes*, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1998.

2. Searles, Scott, *Optical Characterization of Soot Present in a Film Cooled Liquid Rocket Engine Plume*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 2001.

3. Brewer, L.E. and C.C. Limbaugh, *Infrared Band Model Technique for Combustion Diagnostics*, Applied Optics, Vol.11, No. 5, May 1972.

4. Ludwig, C.B. and W. Malkmus, *Update of the Standard Infrared Radiation Model (SIRRM)*. Photon Research Associates, Inc. F04611-82-C-0057. Air Force Rocket Propulsion Laboratory, Air Force Space Technology Center, Space Division, Air Force Systems Command, Edwards Air Force Base, CA, 93523, December 1983.

5. Markarian, P. and R. Kosson, *Standard Infrared Radiation Model (SIRRM-II), Volume I: Algorithm Upgrade Development*. Grumman Corporation. F04611-85-C-0019. Air Force Rocket Propulsion Laboratory, Air Force Space Technology Center, Space Division, Air Force Systems Command, Edwards Air Force Base, CA, 93523, March 1988.

6. Markarian, P. and R. Kosson, *Standard Infrared Radiation Model (SIRRM-II), Volume II: User's Manual*. Grumman Corporation. F04611-85-C-0019. Air Force Rocket Propulsion Laboratory, Air Force Space Technology Center, Space Division, Air Force Systems Command, Edwards Air Force Base, CA, 93523, March 1988.

7. Modest, Michael F. *Radiative Heat Transfer*, McGraw-Hill, Inc., 1993.

8. www.spectraline.com

9. Sivathanu, Yudaya and Rony Joseph, *Fast Infrared Spectrometer: This instrument could be useful in research on fires*, Photonics Tech Briefs, November 2000.

10. *Operation Manual for ES 100 Imaging Spectrometer*, Spectraline, Inc., Vers 1.04, November 2001.

11. *Combustion Species Deconvolution Code*, Arnold Engineering Development Center, Arnold AFB TN, 37389.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
    Ft. Belvoir, VA

2. Dudley Knox Library
    Naval Postgraduate School
    Monterey, CA

3. Curriculum Office, Code 31
    Department of Aeronautics and Astronautics
    Naval Postgraduate School
    Monterey, CA

4. Dr. Christopher M. Brophy, Code AA/Br
    Department of Aeronautics and Astronautics
    Naval Postgraduate School
    Monterey, CA 93943-5000

5. Dr. Jose O. Sinibaldi, Code AA/SJ
    Department of Aeronautics and Astronautics
    Naval Postgraduate School
    Monterey, CA

6. Dr. Ashok Gopinath, Code ME/Gk
    Department of Mechanical Engineering
    Naval Postgraduate School
    Monterey, CA

7. Dr. David W. Netzer, Code AA/Nt
    Department of Aeronautics and Astronautics
    Naval Postgraduate School
    Monterey, CA

8. Dr. Marty Venner
    U.S. Air Force Phillips Laboratory
    Edwards AFB CA

9. Mr. Daniel Brown
    Sverdrup Technology, Inc.
    Arnold AFB TN

10. Mr. Curtis Johnson
    Sierra Engineering
    Carson City, NV

11.   Dr. Jeffrey Muss
       Sierra Engineering
       Carson City, NV

12.   Major Daniel Clepper, USAF
       AFIT/CI
       Wright-Patterson AFB OH

13.   Captain Joshua M. Kutrieb, USAF
       121 Town Farm Road
       North Yarmouth, ME 04097